



TUGAS AKHIR - TE 141599

**KONTROL KECEPATAN MOTOR SEPEDA LISTRIK
MENGUNAKAN *FORCE SENSOR* DAN
ELEKTROMIOGRAFI (EMG)**

Jeffrey Gunawan
NRP 07111440000099

Dosen Pembimbing
Dr. Muhammad Rivai, S.T., M.T.
Ir. Tasripan, M.T.

DEPARTEMEN TEKNIK ELEKTRO
Fakultas Teknologi Elektro
Institut Teknologi Sepuluh Nopember
Surabaya 2018



TUGAS AKHIR - TE 141599

**KONTROL KECEPATAN MOTOR SEPEDA LISTRIK
MENGUNAKAN *FORCE SENSOR* DAN
ELEKTROMIOGRAFI (EMG)**

Jeffrey Gunawan
NRP 07111440000099

Dosen Pembimbing
Dr. Muhammad Rivai, ST., MT.
Ir. Tasripan, MT.

DEPARTEMEN TEKNIK ELEKTRO
Fakultas Teknologi Elektro
Institut Teknologi Sepuluh Nopember
Surabaya 2018



FINAL PROJECT - TE 141599

***SPEED CONTROL SYSTEM OF OF ELECTRICAL
MOTOR IN TORQUE ASSISTED BICYCLE USING
FORCE SENSOR AND ELECTROMYOGRAPHY (EMG)***

Jeffrey Gunawan
NRP 07111440000099

Advisor
Dr. Muhammad Rivai, ST., MT.
Ir. Tasripan, MT.

ELECTRICAL ENGINEERING DEPARTMENT
Faculty of Electrical Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2018

PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul “Kontrol Kecepatan Motor Sepeda Listrik Menggunakan *Force Sensor* dan Elektromiografi (EMG)” adalah benar-benar hasil karya intelektual sendiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diizinkan dan bukan merupakan karya orang lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka. Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai dengan peraturan yang berlaku.

Surabaya, Juli 2018



Jeffrey Gunawan

NRP. 07111440000099

**KONTROL KECEPATAN MOTOR SEPEDA
LISTRIK MENGGUNAKAN *FORCE SENSOR* DAN
ELEKTROMIOGRAFI (EMG)**

TUGAS AKHIR

**Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Sarjana Teknik
Pada**

**Bidang Studi Elektronika
Departemen Teknik Elektro
Fakultas Teknologi Elektro
Institut Teknologi Sepuluh Nopember**

Menyetujui :

Dosen Pembimbing I,



Dosen Pembimbing II,



Dr. Muhammad Rivai, ST., MT.

NIP. 196904261994031003

Ir. Tasripan, M.T.

NIP. 196204181990031004



KONTROL KECEPATAN MOTOR SEPEDA LISTRIK MENGGUNAKAN *FORCE SENSOR* DAN ELEKTROMIOGRAFI (EMG)

Nama : Jeffrey Gunawan
Pembimbing I : Dr. Muhammad Rivai, ST., MT.

ABSTRAK

Torque Assisted Bicycle (TAB) merupakan sebuah sepeda dengan penambahan motor listrik sebagai bantuan bagi penggunanya. Sepeda TAB ini dapat mempermudah pengguna dalam berkendara dan ramah lingkungan karena tidak menghasilkan gas emisi. Namun, sepeda TAB masih belum terlalu diminati di Indonesia. Salah satu penyebabnya ialah penggunaan sensor torsi yang mahal sehingga harga sepeda ini menjadi mahal. Pada penelitian ini, sensor tekanan dan sensor EMG digunakan untuk mengganti sensor tersebut. Data dari dua sensor ini diolah dengan menggunakan mikrokontroler dan sistem *Artificial Neural Network* untuk mengenali data dari dua sensor yang digunakan. Sepeda TAB ini telah dirancang untuk mengenali tingkat aktivitas kaki pengguna sehingga mampu untuk memberikan bantuan kepada pengguna dari motor listrik yang dipasang di sepeda. Daya listrik pada motor ini akan bertambah seiring dengan bertambahnya tekanan pada kaki pengguna. Sinyal dari sensor EMG dan tekanan diolah dengan mikrokontroler Arduino Nano. Pada pengujian statis, subyek mengayuh sepeda pada penyangga selama 10 detik pada 3 tingkat beban yang berbeda dari penggunaan rem. Dari hasil pengujian tersebut, sistem mampu mengenali tiga tingkat beban pada 4 subyek dengan tingkat ketelitian 83,33%. Pada pengujian dinamis, subyek mengayuh sepeda pada 4 tingkat kemiringan permukaan yang berbeda, yakni 0°, 10°, 15°, 20°. Dari hasil pengujian dinamis, sistem mampu mengenali 3 tingkat beban dengan ketelitian rata-rata 75%. Tingkat ketelitian pada dua pengujian tersebut ditinjau dari respon LED yang dihasilkan pada alat.

Kata kunci: ANN, EMG, Sensor Tekanan, Sepeda Listrik, TAB

Halaman ini sengaja dikosongkan

SPEED CONTROL SYSTEM OF ELECTRICAL MOTOR IN TORQUE ASSISTED BICYCLE USING FORCE SENSOR AND ELECTROMYOGRAPHY

Name : Jeffrey Gunawan
1st Advisor : Dr. Muhammad Rivai, ST., MT.

ABSTRACT

Torque Assisted Bicycle is a bike with an addition of an electrical motor as a support for its user. This bike helps its user to ride comfortably without exerting much force and energy when biking and is environmentally friendly as it does not produce any waste. However, TAB is not well known in Indonesia. One of the reason why is because this bike commonly uses a torque sensor, which is an expensive device. Therefore, a pressure sensor and muscle sensor is used to replace the expensive torque sensor. Data from these two sensors will then be processed by a microcontroller and an Artificial Neural Network system in order to classify the data used. This bike is designed to learn and identify its user's foot activity and give a response using the electric motor embedded accordingly. The speed of this electric motor will increase along with the rise in its user's foot activity. Arduino Nano is used to process incoming input from both sensor. Static testing phase is performed when subject performs pedaling motion for 10 seconds on 3 different load which comes from the brake pedal. Results from static testing phase show that system is able to recognize 3 different load on 4 subjects with a 83,33% accuracy. Dynamic testing phase is performed when subject performs pedaling motion in ground with 4 different inclination degree, which are 0°, 10°, 15°, 20°. Result from dynamic testing shows that system is able to recognize 3 different load with a 75% average accuracy. Accuracy rate is measured by looking at the response of the LED in this device.

Key Word: EMG, ANN, TAB, Force Sensor, Electric Bike

Halaman ini sengaja dikosongkan

KATA PENGANTAR

Puji syukur saya panjatkan kepada Tuhan Y.M.E karena berkat dan rahmatNya selama ini penulis dapat menyelesaikan tugas akhir ini.

Penulis juga ingin mengucapkan terimakasih kepada pihak-pihak di bawah ini, yang telah membantu dalam penulisan tugas akhir ini:

- Bapak Dr. Muhammad Rivai, S.T., M.T. selaku dosen pembimbing I dan Bapak Ir. Tasripan, MT. selaku dosen pembimbing II yang memberikan bimbingan dan saran dalam pengerjaan tugas akhir ini.
- Bapak Ir. Dr. Hendra Kusuma, M.Eng.Sc., Ir. Totok Mujiono, M.IKom.,Dr.Eng., Astria Nur Irfansyah, ST.,M.Eng., dan Muhammad Attamimi, B.Eng, M.Eng, PhD. selaku dosen penguji saat sidang tugas akhir yang sudah memberikan
- Seluruh keluarga saya yang sudah memberikan semangat dan doa agar tugas akhir ini dapat berjalan dengan baik.
- Teman-teman seperjuangan asisten laboratorium elektronika yang sudah memberi dukungan.

Akhir kata, seperti kata pepatah, “tidak ada gading yang tak retak”, penulis sepenuhnya menyadari bahwa tugas akhir ini masih jauh dari sempurna, dan mohon maaf yang sebesar-besarnya apabila masih terdapat kesalahan dalam penulisan tugas akhir ini. Penulis berharap bahwa tugas akhir ini dapat memberikan manfaat bagi rekan-rekan yang akan melanjutkan riset di bidang elektronika.

Surabaya, Juni 2018

Penulis

Halaman ini sengaja dikosongkan

DAFTAR ISI

ABSTRAK	i
ABSTRACT	iii
KATA PENGANTAR	v
DAFTAR ISI	vii
DAFTAR GAMBAR	ix
DAFTAR TABEL.....	xi
BAB I	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah.....	2
1.3 Tujuan Penelitian	2
1.4 Batasan Masalah	2
1.5 Metodologi Penelitian	2
1.6 Sistematika Penulisan.....	4
1.7 Relevansi	4
BAB II.....	5
1.1 Elektromiografi	5
1.2 Elektroda.....	7
1.2.1 <i>Surface Gelled Electrode</i>	8
1.3 Sensor Tekanan	9
1.4 Arduino	10
1.5 Artificial Neural Network (ANN).....	11
1.6 Kontroler PID.....	14
1.7 <i>Moving Average Filter</i>	15
1.8 Motor DC.....	16
BAB III.....	19
3.1 Diagram Blok Sistem	19
3.2 Perancangan Perangkat Keras.....	21
3.2.1 Perancangan Sensor EMG	21
3.2.2 Perancangan Sensor Tekanan	23
3.2.3 Arduino Nano.....	25
3.3. Perancangan Perangkat Lunak	27
3.3.1 Perancangan Perangkat Lunak pada Komputer	27
3.3.2 Perancangan Perangkat Lunak pada Arduino	32

3.4 Perancangan Mekanik TAB	34
BAB IV	35
4.1 Realisasi Desain TAB	35
4.2 Pengujian Perangkat Keras	36
4.2.1 Pengujian Sensor EMG pada Tangan.....	36
4.2.2 Pengujian Sensor EMG pada Gerak <i>Cycling</i>	38
4.2.3 Pengujian Sensor Tekanan pada Gerak <i>Cycling</i>	41
4.2.4 Pengujian Sensor EMG dan Tekanan pada Gerak <i>Cycling</i>	42
4.2.5 Pengujian PWM pada Motor DC	44
4.3 Pengujian Perangkat Lunak.....	45
4.3.1 Pengujian Filter <i>Moving Average</i>	45
4.3.2 Pengujian Sistem Pembelajaran ANN	46
4.3.3 Pengujian Propagasi <i>Feedforward</i> pada Arduino.....	51
4.4 Pengujian Akhir pada Kondisi Statis	52
4.5 Pengujian Akhir pada Kondisi Dinamis	54
BAB V	57
5.1 Kesimpulan	57
5.2 Saran	57
DAFTAR PUSTAKA.....	59
LAMPIRAN A	61
LAMPIRAN B	73
BIODATA PENULIS.....	81

DAFTAR GAMBAR

Gambar 2. 1	Sinyal EMG yang belum Diolah.....	5
Gambar 2. 2	Penempatan Elektroda pada Otot.....	6
Gambar 2. 3	Ilustrasi Otot <i>Vastus Lateralis</i>	6
Gambar 2. 4	Ilustrasi Otot <i>Gastrocnemius</i> dan <i>Soleus</i>	7
Gambar 2. 5	Elektroda <i>Gelled</i>	8
Gambar 2. 6	<i>Force Sensitive Resistor</i>	9
Gambar 2. 7	Ilustrasi Lapisan pada FSR	10
Gambar 2. 8	Arduino Nano.....	10
Gambar 2. 9	Ilustrasi Neuron pada Tubuh Manusia	12
Gambar 2.10	Model Matematis Neuron.....	12
Gambar 2.11	Grafik Karakteristik Fungsi Transfer	13
Gambar 2.12	Ilustrasi Multilayer Neural Network	14
Gambar 2.13	Diagram Blok Kontroler PID.....	15
Gambar 2.14	Contoh Sinyal dengan MA Filter	16
Gambar 2.15	Konstruksi dalam Motor DC.....	16
Gambar 2.16	Motor DC 775	17
Gambar 3.1	Diagram Blok Sistem	19
Gambar 3.2	Skematik Rangkaian Akuisisi Sinyal EMG.....	21
Gambar 3.3	Plot Diagram <i>Frequency vs Gain</i>	22
Gambar 3.4	Grafik Kerja FSR	23
Gambar 3.5	Rangkaian Sensor Tekanan Kondisi Setengah Beban.....	24
Gambar 3.6	Rangkaian Sensor Tekanan Kondisi Tanpa Beban.....	24
Gambar 3.7	Ilustrasi Titik Tekanan pada Telapak Kaki.....	25
Gambar 3.8	Skema Pengkabelan Mikrokontroler	26
Gambar 3.9	Diagram ANN yang akan Direalisasikan	27
Gambar 3.10	<i>Flowchart</i> Program ANN	31
Gambar 3.11	<i>Flowchart</i> Program Arduino.....	33
Gambar 3.12	Desain Mekanik TAB.....	34
Gambar 4.1	Realisasi Desain Sepeda	35
Gambar 4.2	Grafik Perubahan Frekuensi Terhadap Gain Sensor EMG.....	36
Gambar 4.3	Respon Sensor terhadap Kontraksi Otot Tangan	37
Gambar 4.4	Peletakan Elektroda pada Tangan	37
Gambar 4.5	Peletakan Elektroda pada Otot <i>Gastrocnemius</i> dan <i>Soleus</i>	39

Gambar 4.6	Peletakan Elektroda pada Otot <i>Vastus Lateralis</i>	39
Gambar 4.7	Pengukuran EMG pada Otot <i>Gastrocnemius</i> dan <i>Soleus</i>	40
Gambar 4.8	Pengukuran EMG pada Otot <i>Vastus Lateralis</i>	40
Gambar 4.9	Sensor Tekanan pada Sandal	41
Gambar 4.10	Pengukuran Sensor Tekanan pada titik <i>Anterior</i>	41
Gambar 4.11	Pengukuran Sensor Tekanan pada titik <i>Medial</i>	42
Gambar 4.12	Pengukuran Sensor Tekanan pada titik <i>Posterior</i>	42
Gambar 4.13	Pengukuran Sensor EMG dan Tekanan Kondisi Tanpa Beban	43
Gambar 4.14	Pengukuran Sensor EMG dan Tekanan Kondisi Setengah Beban.....	43
Gambar 4.15	Pengukuran Sensor EMG dan Tekanan Kondisi Beban Penuh	43
Gambar 4.16	Grafik Perbandingan Nilai <i>analogWrite</i> dibandingkan RPM.....	44
Gambar 4.17	Pengujian Filter MAV	45
Gambar 4.18	Data Pembelajaran Sebelum Normalisasi.....	47
Gambar 4.19	Data Pembelajaran Sesudah Normalisasi	47
Gambar 4.20	Perubahan Nilai <i>Weight</i> 1 terhadap Iterasi ANN	48
Gambar 4.21	Perubahan Nilai <i>Bias</i> 1 terhadap Iterasi ANN	48
Gambar 4.22	Perubahan Nilai <i>Weight</i> 2 terhadap Iterasi ANN	49
Gambar 4.23	Perubahan Nilai <i>Bias</i> 2 terhadap Iterasi ANN	49
Gambar 4.24	Subyek pada Pengujian Statis	52
Gambar 4.25	Sistem Perangkat Keras Akhir	53

DAFTAR TABEL

Tabel 3.1	Penggunaan Pin Arduino	26
Tabel 4.1	Data Pembelajaran ANN A.....	46
Tabel 4.2	Data Pembelajaran ANN B.....	46
Tabel 4.3	Data SSE A	50
Tabel 4.4	Data SSE B.....	51
Tabel 4.5	Data Pengujian ANN pada Arduino	51
Tabel 4.6	Data Pengujian Statis	53
Tabel 4.7	Data Pengujian Dinamis	54

Halaman ini sengaja dikosongkan

BAB I

PENDAHULUAN

1.1 Latar Belakang

Semakin bertambahnya jumlah kendaraan bermotor di Indonesia merupakan salah satu isu yang terus diperhatikan oleh pemerintah. Menurut data dari BPS (Badan Pusat Statistik), di tahun 2016, terdapat 105.150.082 sepeda motor yang beredar di Indonesia. Sepeda motor lebih diminati oleh masyarakat karena tidak pengguna tidak perlu menggunakan tenaga yang cukup besar untuk menggerakkan sepeda motor. Sepeda merupakan salah satu alternatif moda transportasi yang dapat digunakan oleh masyarakat. Namun, karena iklim Indonesia yang cenderung panas dan kondisi geografis Indonesia yang bervariasi, sepeda kurang diminati oleh masyarakat. Untuk mengatasi kekurangan ini, dikembangkanlah sebuah *Torque Assisted Bicycle* (TAB).

TAB merupakan sebuah sepeda dengan fitur berupa motor listrik yang dapat menghasilkan torsi tambahan ketika diperlukan. Fitur ini berguna untuk membantu pengguna ketika pengguna berkendara pada medan yang terjal sehingga memerlukan tenaga lebih besar untuk mengayuh sepeda. Pada saat medan terjal, motor listrik akan menyala untuk memberi bantuan torsi sehingga pengguna tidak perlu mengayuh terlalu keras, dan sebaliknya pada kondisi medan yang landai, motor listrik akan dimatikan sehingga pengguna menggunakan tenaganya sendiri untuk mengayuh pedal sepeda.

Salah satu masalah terkait dengan TAB ini ialah bahwa harga sepeda jenis ini masih mahal. Salah satu faktor yang menyebabkan tingginya harga sepeda ini ialah sensor torsi yang disematkan untuk fungsi bantuan motor [1]. Oleh sebab itu, dibutuhkan sensor alternatif untuk menjalankan fungsi bantuan motor ini.

Pada tugas akhir ini, akan dirancang sebuah mekanisme sistem kontrol motor listrik pada TAB melalui masukan dari 2 sensor, yakni sensor Elektromiografi (EMG) dan sensor tekanan. Sensor EMG berfungsi untuk mendeteksi aktivitas otot dari otot kaki yang digunakan untuk mengayuh pedal, sedangkan sensor tekanan berfungsi mendeteksi berapa besar tekanan yang diberikan kaki pengguna pada pedal. Masukan

ini kemudian akan diolah untuk mengekstraksi fitur yang terdapat pada sinyal EMG, kemudian dimasukkan pada sebuah sistem kecerdasan buatan untuk menentukan apakah diperlukan bantuan motor listrik untuk membantu pengguna sepeda. Dengan menggunakan sistem kontrol ini, pengguna akan menjadi lebih nyaman dalam menggunakan sepeda karena tidak perlu mengayuh terlalu berat pada saat berada dalam medan yang terjal. Dengan dikembangkannya sistem ini, diharapkan minat masyarakat terhadap sepeda listrik TAB menjadi semakin besar.

1.2 Perumusan Masalah

Berdasarkan latar belakang di atas, dapat dirumuskan beberapa masalah, antara lain:

1. Bagaimana penempatan sensor yang optimal.
2. Bagaimana mengolah sinyal EMG agar siap digunakan oleh mikrokontroller.
3. Sistem kecerdasan apa yang digunakan untuk menghasilkan keputusan.

1.3 Tujuan Penelitian

Penelitian pada tugas akhir ini bertujuan sebagai berikut:

1. Mengimplementasikan sensor tekanan dan EMG pada beberapa titik uji yang berbeda.
2. Mengimplementasikan filter analog dari rangkaian dan rektifikasi agar sinyal dapat dibaca melalui ADC dari mikrokontroler.
3. Merancang sistem *Artificial Neural Network* (ANN) sebagai metode pembelajaran untuk mengenali masukan dari sensor.

1.4 Batasan Masalah

Batasan masalah dari tugas akhir ini adalah sebagai berikut:

1. Sepeda dikayuh pada kecepatan konstan.
2. Subyek berumur 20-23 tahun, sehat secara jasmani dan rohani, dan tidak memiliki cacat fisik apapun.

1.5 Metodologi Penelitian

Dalam penyelesaian tugas akhir ini digunakan metodologi sebagai berikut:

1. Studi literatur

Pada tahap ini dilakukan pengumpulan dasar teori yang dapat menunjang penulisan tugas akhir. Berikut merupakan studi literatur yang dilakukan:

- a. Studi mengenai anatomi otot manusia, terutama otot kaki.
- b. Studi mengenai penempatan elektroda yang optimal pada otot kaki.
- c. Studi mengenai rangkaian akuisisi sinyal EMG dan karakteristik sinyal EMG.
- d. Studi mengenai rangkaian sensor tekanan.
- e. Studi mengenai pengolahan data menggunakan Arduino Nano.
- f. Studi mengenai sistem pengaturan kecepatan pada motor listrik menggunakan PWM.

Dasar teori ini dapat diambil dari buku, jurnal, *proceeding*, serta artikel dari internet.

2. Perancangan Sistem *Hardware*

Setelah mempelajari literature yang ada, selanjutnya akan dilakukan perancangan sistem. Sistem yang akan dirancang meliputi perancangan sensor EMG dan sensor tekanan.

3. Perancangan Sistem *Software*

Pada tahap ini dilakukan perancangan serta pengujian *software* yang digunakan untuk mengolah data dari sensor dan menampilkan data tersebut secara visual melalui Arduino IDE.

4. Pengujian Keseluruhan Sistem

Pada tahap ini dilakukan pengujian sistem setelah melakukan perekaman data terlebih dahulu. Kemudian, sistem diuji secara statis dan dinamis.

5. Penulisan Laporan Tugas Akhir

Tahap ini dilakukan saat sistem telah selesai diuji dan data yang diinginkan sudah didapatkan.

1.6 Sistematika Penulisan

Laporan tugas akhir ini terdiri dari lima bab dengan sistematika penulisan sebagai berikut:

- Bab 1: Pendahuluan

Bab ini meliputi latar belakang, perumusan masalah, tujuan penelitian, batasan masalah, metodologi, sistematika penulisan, dan relevansi.

- Bab 2: Dasar Teori

Bab ini menjelaskan tentang dasar-dasar teori yang dibutuhkan dalam pengerjaan tugas akhir ini, yang meliputi teori dasar mengenai sensor elektromiografi, sensor FSR, mikrokontroler Arduino, dan sistem kontrol kecepatan motor.

- Bab 3: Perancangan Alat

Bab ini menjelaskan tentang perencanaan sistem baik perangkat keras (*hardware*) maupun perangkat lunak (*software*) untuk mengolah data dari sensor elektromiografi dan sensor tekanan.

- Bab 4: Pengujian Alat

Bab ini menjelaskan tentang hasil yang didapat dari tiap sensor, kemudian menguji sistem pembelajaran ANN, serta menguji gerak motor sebagai respon dari perintah yang diberikan dari mikrokontroler.

- Bab 5: Penutup

Bab ini menjelaskan tentang kesimpulan, meliputi kekurangan-kekurangan pada kerja alat dari hasil analisa serta saran untuk pengembangan ke depan.

1.7 Relevansi

Hasil dari tugas akhir ini diharapkan dapat memberikan manfaat sebagai berikut:

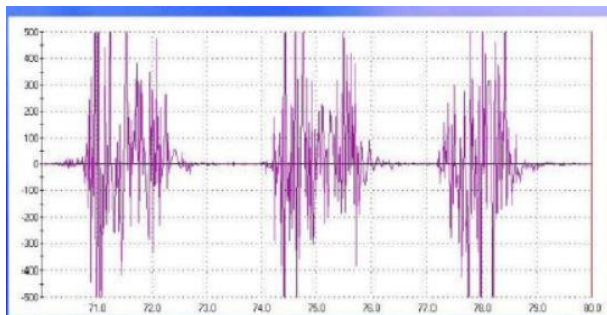
1. Dapat mengidentifikasi kapan pengguna sepeda memerlukan bantuan dari motor listrik dalam beberapa tingkat beban yang berbeda.
2. Dapat mengatur kecepatan motor listrik berdasarkan tingkat aktivitas kaki pengguna

BAB II

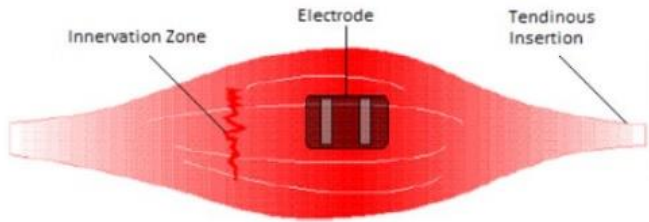
DASAR TEORI

1.1 Elektromiografi

Elektromiografi merupakan teknik yang digunakan untuk tujuan deteksi, analisa, dan pemanfaatan sinyal listrik yang berasal dari otot tubuh manusia. Sinyal listrik ini dihasilkan ketika terjadi kontraksi pada otot dan sebuah arus listrik kecil dihasilkan oleh terjadinya pertukaran ion pada membran otot[2]. Sinyal EMG dapat memberikan informasi mengenai aktivitas otot tubuh manusia dan amplitudo sinyal EMG meningkat sebanding dengan kontraksi pada otot yang dituju [3], dan sinyal ini dapat dideteksi menggunakan elektroda tanam maupun elektroda di permukaan kulit. Untuk riset ini digunakan elektroda tempel di permukaan kulit karena pengguna hanya perlu menempelkan elektroda pada saat mengendarai sepeda saja. Dalam merekam sinyal EMG, penempatan elektroda perlu diperhatikan agar sinyal yang direkam baik dan dapat dianalisa. Elektroda hendaknya diletakkan diantara unit motorik dan *tendinous insertion* dari otot, yakni sepanjang garis longitudinal dari otot. Contoh sinyal EMG yang belum mengalami proses pengondisian sinyal dapat dilihat pada gambar 2.1.

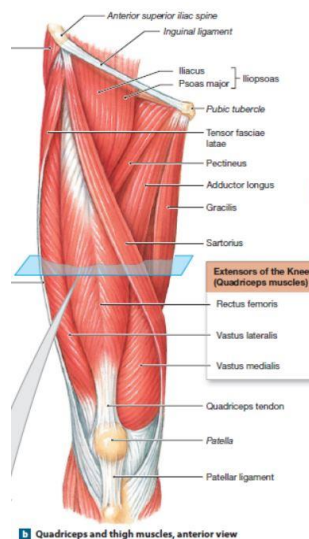


Gambar 2.1 Sinyal EMG yang belum Diolah

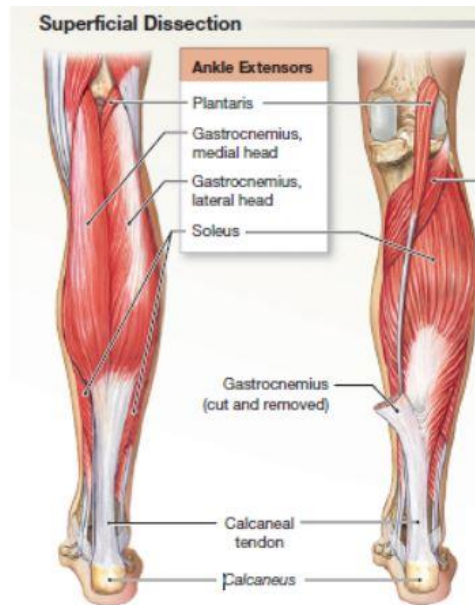


Gambar 2.2 Ilustrasi Penempatan Elektroda

Gambar 2.2 menunjukkan titik penempatan elektroda yang disarankan dalam perekaman sinyal EMG. Pada penelitian ini, otot kaki yang digunakan ialah otot *Vastus Lateralis*, *Gastrocnemius* dan *soleus* [4]. Adapun ketiga otot ini dipilih karena merupakan otot yang dominan ketika kaki mengayuh sepeda. Berikut merupakan ilustrasi otot yang digunakan[5]:



Gambar 2.3 Ilustrasi otot *Vastus Lateralis*



Gambar 2.4 Ilustrasi Otot *Gastrocnemius* dan *Soleus*

Namun, banyak faktor yang memengaruhi sinyal EMG, seperti faktor biologis dari masing-masing pengguna (jenis kulit, kandungan lemak pada kulit, dan lain-lain), jenis otot yang akan dianalisa, antarmuka elektroda dan kulit, penempatan elektroda, dan lain-lain. Sehingga, diperlukan pengolahan sinyal terlebih dahulu sebelum sinyal EMG dimasukkan ke sebuah mikrokontroller.

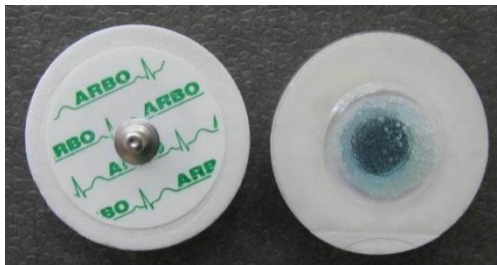
1.2 Elektroda

Aktivitas bioelektrik pada otot tubuh manusia dapat dideteksi dengan menggunakan elektroda. Dalam melakukan pengukuran EMG, terdapat istilah elektroda referensi dan elektroda aktif. Elektroda referensi diletakkan pada bagian bertulang yang jauh dari otot yang hendak diukur sementara elektroda aktif diletakkan pada bagian tengah otot yang hendak diukur. Dalam melakukan pengukuran EMG, Terdapat 2 tipe dari elektroda yang digunakan pada pengukuran EMG, yakni yang terletak pada permukaan (*surface electrode*) atau yang dimasukkan pada otot (*inserted electrode*).

1.2.1 Surface Gelled Electrode

Elektroda jenis ini diletakkan pada permukaan kulit tubuh manusia, sehingga tidak memerlukan prosedur yang bersifat invasif. Elektroda ini mampu membentuk kesetimbangan antara permukaan elektroda dan permukaan kulit melalui konduksi elektrolitik, sehingga arus tersebut dapat mengalir menuju elektroda[2]. Kelebihan elektroda jenis ini ialah prosedur penggunaannya yang lebih mudah dan tidak memerlukan pengawasan medis yang ketat karena tidak perlunya operasi untuk penanaman elektroda. Selain itu, elektroda ini dapat langsung dibuang setelah digunakan sehingga memudahkan pengguna dalam proses pemasangan dan pelepasan. Namun, kekurangan elektroda jenis ini ialah memerlukan posisi yang stabil dan tidak bergerak ketika dipasang pada kulit untuk menghindari *noise* ketika otot digerakkan sehingga diperlukan persiapan pada kulit untuk menghindari *noise* tersebut, seperti misalnya membersihkan permukaan kulit dengan cairan alkohol.

Elektroda jenis ini mengandung suatu bahan gel elektrolitik sebagai antarmuka antara kulit dan elektroda. Reaksi reduksi dan oksidasi terjadi pada persambungan logam di elektroda. Bahan yang paling umum digunakan untuk elektroda jenis ini ialah bahan Ag-AgCl pada bagian logam dari elektroda. Lapisan AgCl ini memungkinkan arus dari otot untuk lewat secara lebih leluasa pada persambungan antara elektrolit dan elektroda. Hal ini memungkinkan hasil pengukuran yang lebih baik karena *noise* pada pengukuran tidak banyak seperti halnya pada elektroda metalik lain (Contoh: Ag)[2].



Gambar 2.5 Elektroda *Gelled*

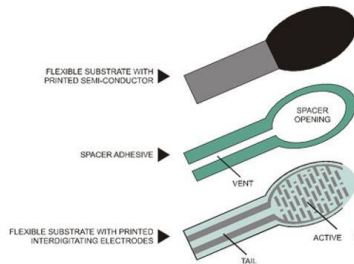
1.3 Sensor Tekanan

Sensor tekanan yang digunakan merupakan sebuah *force sensitive resistor* yang berbentuk lingkaran pada ujungnya dengan daerah *sensing* 0.5 inch. Resistor ini akan mengalami perubahan nilai hambatan tergantung dari seberapa besar tekanan yang diberikan pada daerah *sensing*. Semakin besar tekanan yang diberikan, maka resistansi pada sensor akan menjadi semakin rendah. Ketika tidak ada tekanan yang diberikan pada sensor, maka nilai resistansinya menjadi lebih besar dari 1M Ω . Sensor ini dapat mendeteksi nilai tekanan pada rentang 100g hingga 10kg.

Gambar 2.6 menunjukkan bentuk fisik dari FSR. Sensor ini terbuat dari bahan polimer konduktif, yang resistansinya berubah tergantung dari seberapa besar tekanan yang diberikan pada permukaannya. Selain itu, sensor ini terdiri dari beberapa lapisan yang memiliki karakteristik yang berbeda satu dengan yang lain. Sensor ini terdiri dari 3 lapisan, yaitu lapisan resistif, lapisan perekat, dan lapisan konduktif yang disusun secara bertumpuk, seperti yang terlihat pada gambar 2.7. Sehingga, ketika suatu gaya diberikan pada sensor ini, lapisan konduktif akan bersentuhan dengan lapisan resistif sehingga akan timbul nilai resistansi. Jika tidak ada gaya yang diberikan, maka seolah-olah sensor bersifat seperti rangkaian *open loop* yang nilai resistansinya tak berhingga.



Gambar 2.6 *Force Sensitive Resistor*



Gambar 2.7 Ilustrasi Lapisan pada FSR

Sensor ini akan diletakkan pada telapak kaki pengguna untuk mengetahui besar tekanan yang diberikan oleh telapak kaki pada pedal. Ketika pengguna mengayuh sepeda pada medan yang landai, maka tekanan yang diberikan pada sensor kecil. Namun, ketika pengguna mengayuh sepeda pada medan yang terjal, tekanan yang diberikan pada sensor menjadi besar. Perbedaan tekanan yang menyebabkan timbulnya perbedaan resistansi sensor inilah yang akan dideteksi oleh sistem untuk mengenali kapan diperlukan bantuan dari motor listrik.

1.4 Arduino

Arduino merupakan perangkat elektronik *open source* yang menggunakan perangkat keras dan perangkat lunak yang mudah untuk digunakan. Arduino mampu untuk membaca masukan, seperti cahaya dari sebuah sensor, tombol, maupun pesan singkat, dan merubah masukan tersebut menjadi suatu bentuk keluaran, seperti menggerakkan motor, menyalakan LED, dan sebagainya. Hal ini dapat dilakukan dengan mengirimkan serangkaian instruksi ke mikrokontroler pada papan Arduino dengan menggunakan *Arduino IDE*, yaitu perangkat pemrogram untuk papan Arduino yang dapat diunduh dari internet.



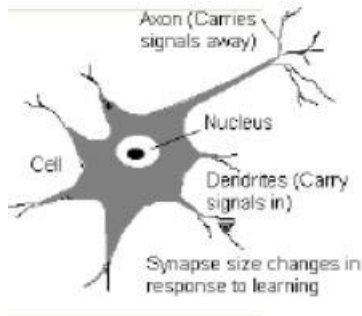
Gambar 2.8 Arduino Nano

Arduino Nano merupakan sebuah mikrokontroler yang masih menjadi bagian dari keluarga mikrokontroler Arduino, namun ukurannya lebih kecil dan masih menggunakan IC ATmega328P. Perbedaan yang dapat dijumpai ialah tidak adanya DC Jack, melainkan menggunakan kabel Mini-B USB untuk pemrogramannya. Arduino Nano dapat dinyalakan dengan menggunakan kabel Mini-B USB, Tegangan 6-20V dari catu daya eksternal pada pin 30, ataupun Tegangan 5V dari catu daya eksternal pada pin 27. ATmega328P memiliki 32KB memori flash dan 2KB SRAM. Arduino Nano memiliki 14 pin digital yang dapat digunakan sebagai masukan maupun keluaran, dengan tegangan operasi sebesar 5V. Tiap pin dapat menerima maximum 40mA dan memiliki resistor *pull-up* sebesar 20-50kOhm. Arduino ini juga memiliki 8 pin masukan analog dan tiap pin menyediakan resolusi sebesar 10bit. Arduino Nano menggunakan skema komunikasi UART TTL(5V) untuk berkomunikasi dengan komputer, Arduino lain, ataupun mikrokontroler lain.

1.5 Artificial Neural Network (ANN)

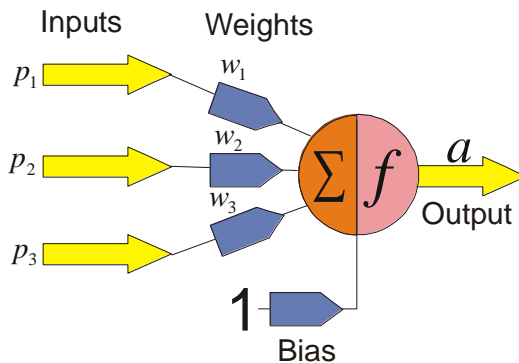
Jaringan Saraf Tiruan / Artificial Neural network (ANN) merupakan salah satu komponen penting dalam *artificial intelligence* (AI). Hal ini telah bertahun-tahun dipelajari dengan harapan sebuah mesin mampu mencapai kinerja seperti otak manusia. ANN merupakan model komputasional yang berdasarkan struktur dan fungsi dari jaringan saraf biologis. Informasi yang mengalir melalui jaringan turut memengaruhi struktur dari ANN karena jaringan saraf berubah, atau dapat dikatakan belajar, melalui masukan dan keluaran. Artificial neural network adalah sejumlah besar prosesor yang terdistribusi secara paralel dan terdiri dari unit pemrosesan sederhana dan masing-masing unit memiliki kecenderungan untuk menyimpan pengetahuan yang dialami dan dapat digunakan kembali. ANN terinspirasi dari sebuah *Neuron*, unit penyusun sistem saraf manusia.

Suatu neuron memiliki banyak jalur masukan dan satu jalur keluaran, seperti dilihat pada gambar 2.9. Pada gambar tersebut, dapat dilihat ilustrasi neuron pada tubuh manusia. Jalur masukan yang disebut dendrit membawa sinyal masuk ke neuron dan suatu jalur keluaran yang disebut akson membawa sinyal keluar dari neuron. Begitu pula halnya pada ANN, terdapat masukan dan keluaran pada tiap neuron.



Gambar 2.9 Ilustrasi Neuron pada Tubuh Manusia

Keluaran dari neuron tersebut dapat berupa kondisi tereksitasi atau tidak tereksitasi. Sinyal masukan dari neuron lain menentukan kondisi keluaran dari neuron tersebut. Keluaran yang dihasilkan neuron juga mengalami atenuasi / penundaan pada *synapse*, yang merupakan sebuah persambungan dari neuron. Model matematis dari sebuah neuron dapat dilihat pada gambar 2.10. Pada jalur di gambar 2.10, misalkan terdapat 3 jalur masukan, dan terdapat “bobot”, disebut *weight* dan *bias*, serta f (fungsi transfer).

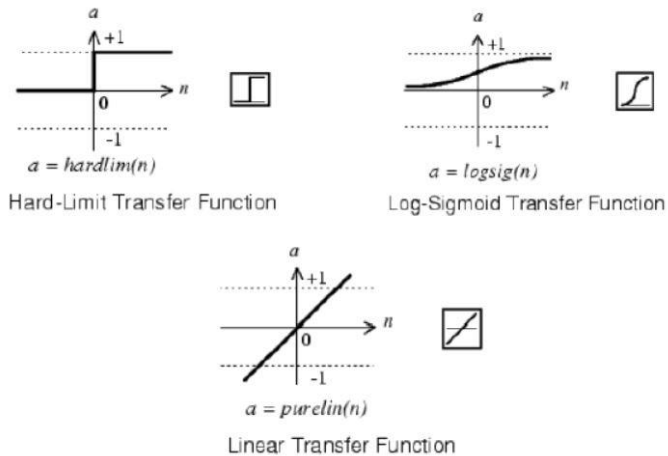


Gambar 2.10. Model Matematis dari Neuron

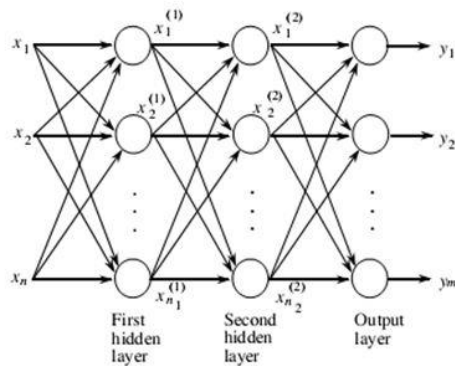
$$a = f(p_1w_1 + p_2w_2 + p_3w_3 + b) = f\left(\sum p_iw_i + b\right) \quad (2.1)$$

Model matematis seperti gambar 2.10 dapat dirumuskan seperti persamaan 2.1. Terdapat beberapa macam fungsi transfer, seperti *hard limit function*, *log sigmoid function*, dan *linear function*. Tiap fungsi tersebut memiliki karakteristik untuk tersendiri. Grafik karakteristik dari tiap fungsi tersebut ditunjukkan pada gambar 2.11. Tiap fungsi transfer yang berbeda dapat digunakan untuk kebutuhan yang berbeda pula.

Terdapat beberapa jenis arsitektur dari ANN. Namun, pada riset ini digunakan *multilayer neural network*. Tiap *layer* / lapisan dari sebuah jaringan multilayer menjalankan tugas yang berbeda. Lapisan *input layer* merupakan masukan dari ANN. Lapisan yang menghasilkan keluaran disebut *output layer*. Lapisan *hidden layer* menjalankan fungsi matematis untuk perhitungan pada tiap neuron didalam ANN.



Gambar 2.11 Grafik Karakteristik Fungsi Transfer



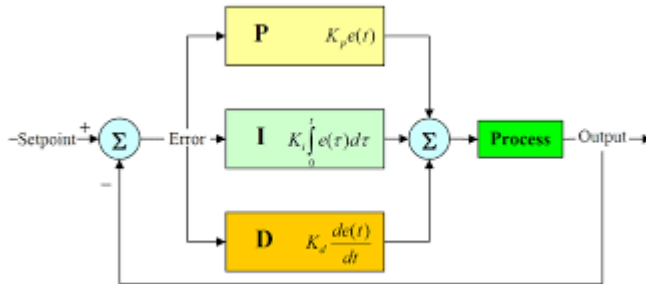
Gambar 2.12 Ilustrasi Multilayer Neural Network

Pada ilustrasi ANN dengan beberapa layer di gambar 2.12, terdapat 1 *output layer* (layer 3) dan 2 *hidden layer* (layer 1 dan 2). Jaringan multilayer merupakan salah satu jaringan yang kuat. Sebagai contoh, sebuah jaringan memiliki 2 lapisan dengan lapisan pertama memiliki fungsi transfer *sigmoid*, sedangkan lapisan kedua memiliki fungsi transfer *linear*, dapat dilatih untuk memperkirakan banyak fungsi secara cukup baik.

ANN adalah metode yang sering digunakan dalam berbagai riset, seperti pengenalan suara burung [6], pendeteksian kualitas kesegaran daging [7], dan hidung elektronik [8] [9] [10].

1.6 Kontroler PID

Kontrol PID merupakan salah satu sistem kontrol yang sering digunakan pada aplikasi industri. Kegunaan kontroler PID ini terletak pada sifatnya yang dapat diaplikasikan pada banyak sistem, bahkan ketika model matematis dari sistem tersebut tidak diketahui sehingga menyebabkan metode desain analitik tidak dapat digunakan[11]. Kontroler ini akan menggunakan nilai *error* yang didapatkan dari selisih antara keluaran sistem (*plant*) dan nilai ideal yang diinginkan (*setpoint*) yang sudah ditetapkan sebelumnya untuk mengurangi *error* tersebut. Terdapat 3 elemen dalam kontroler ini, yaitu *Proportional* (P), *Integral* (I), dan



Gambar 2.13 Diagram Blok Kontroler PID

Derivative (D), dengan tiap elemen memiliki respon berbeda terhadap *plant* sehingga perlu diperhitungkan secara teliti dalam desain kontroler.

Pada gambar 2.13, nilai *error* adalah selisih antara keluaran ideal dan keluaran riil. Sinyal keluaran dari kontroler sendiri dapat dirumuskan sebagai berikut dalam domain waktu:

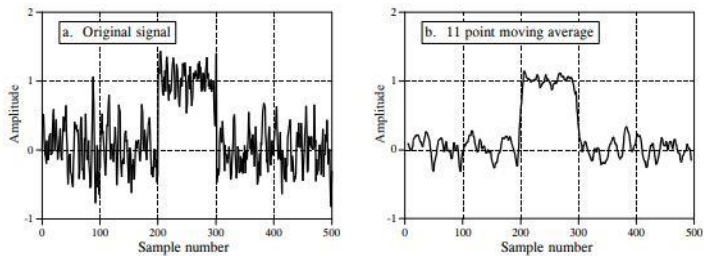
$$u(t) = K_p * e(t) + K_i * \int e(t)dt + K_d \frac{de}{dt} \quad (2.2)$$

K_p , K_i , dan K_d merupakan konstanta penguatan proporsional, penguatan integral, dan penguatan derivative. Tiap elemen konstanta tersebut memiliki respon yang berbeda terhadap sistem sehingga diperlukan *tuning* secara manual untuk mendapatkan nilai yang optimal. Kemudian, hasil $u(t)$ digunakan sebagai nilai koreksi agar sistem semakin mendekati nilai *setpoint* yang diinginkan.

1.7 Moving Average Filter

Moving Average (MA) *Filter* adalah salah satu jenis filter digital yang paling sering digunakan karena filter jenis ini adalah filter yang mudah untuk dipahami dan digunakan. Filter MA ini cocok digunakan untuk mengurangi gangguan *noise* yang tidak diinginkan dalam suatu hasil pengukuran. Karakteristik filter ini ialah memiliki respon domain waktu yang bagus namun memiliki respon domain frekuensi yang buruk. Berikut merupakan rumus yang digunakan dalam filter MA ini:

$$y[i] = \frac{1}{M} \sum_{j=0}^{M-1} x[i + j] \quad (2.3)$$



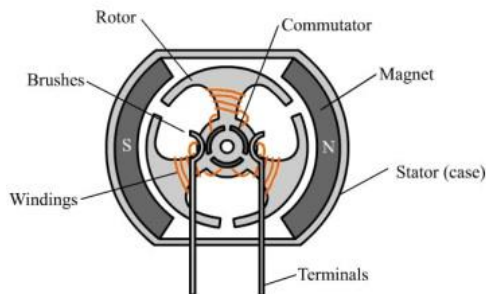
Gambar 2.14 Contoh Sinyal dengan MA Filter

Dari persamaan 2.3, $x[]$ merupakan sinyal masukan, $y[]$ merupakan sinyal keluaran, dan M adalah jumlah sampel data yang digunakan. Filter ini bekerja dengan cara mengambil sejumlah sampel data, yakni M , kemudian mencari nilai rata-rata dari sejumlah sampel data tersebut, dan nilai tersebut merupakan keluaran filter untuk sampel data tersebut. Dengan semakin bertambahnya banyak sampel data (parameter M), ‘kehalusan’ dari respon keluaran meningkat, namun respon transisi pada data semakin tumpul.

Gambar 2.14 menunjukkan sebuah sinyal sebelum dan sesudah penggunaan filter MA[12]. Dari contoh tersebut, dapat dilihat bahwa penggunaan filter MA ini mampu menghilangkan *noise* yang merusak hasil pengukuran tersebut.

1.8 Motor DC

Motor DC merupakan sebuah motor yang bergerak dengan mengubah energi listrik menjadi gerakan. DC berarti *Direct Current*, sehingga motor ini dapat disebut motor arus searah. Motor jenis ini



Gambar 2.15 Konstruksi dalam Motor DC

memiliki 2 *terminal* dan memerlukan tegangan searah untuk menggerakkannya. Motor jenis ini biasa dijumpai pada kipas angin, bor listrik, dan berbagai perangkat lain.

Secara garis besar, terdapat 2 jenis motor DC yang sering digunakan, yakni motor DC *brushed* dan *brushless*. Pada riset ini, yang digunakan ialah motor DC *brushed*. Pada motor jenis ini, terdapat 2 bagian di dalam motor, yakni rotor dan stator. Stator merupakan bagian motor yang diam sedangkan rotor merupakan bagian yang berputar, atau bisa disebut *armature*. Salah satu keunggulan dari motor DC jenis ini ialah desain yang sederhana dan harga yang murah. Namun, kelemahannya ialah efisiensinya rendah dan tidak cocok apabila tegangan kerja yang digunakan tinggi.

Prinsip kerja dari motor ini cukup sederhana. Tegangan yang diberikan pada kedua *terminal* dari motor akan melewati *brush* kemudian mengalir menuju komutator yang berfungsi mengubah polaritas arus listrik. Kemudian, arus ini mengalir pada konduktor electromagnet dan menyebabkan rotor berputar karena adanya medan magnet yang berubah pada rotor. Arah putaran rotor ini mengikuti aturan *Fleming*.

Pada tugas akhir ini, motor DC yang digunakan yaitu motor DC 775. Motor ini digunakan karena memiliki torsi yang cukup besar dan bentuk yang cukup kompak sehingga mudah untuk dipasang pada sepeda. Penggerak yang digunakan akan dipasang pada as motor untuk memutar ban sepeda. Motor ini mampu bekerja pada tegangan kerja 12-36V dan mampu berputar pada 3500-9000RPM, tergantung dari tegangan yang diberikan pada motor. Gambar 2.16 menunjukkan bentuk fisik motor ini.



Gambar 2.16 Motor DC 775

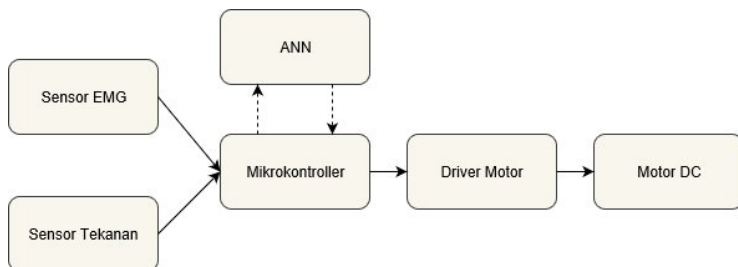
Halaman ini sengaja dikosongkan

BAB III PERANCANGAN SISTEM

Bab ini menjelaskan perancangan sistem pada alat secara keseluruhan, meliputi perangkat keras dan perangkat lunak. Perangkat keras yang digunakan meliputi sensor EMG, sensor tekanan, mikrokontroller Arduino Nano, motor DC, serta rangkaian *driver* motor DC. Perangkat lunak meliputi program untuk proses *neural network* serta program pembacaan nilai sensor pada Arduino.

3.1 Diagram Blok Sistem

Gambar 3.1 menunjukkan cara kerja sistem dari masukan sensor hingga keluaran pada aktuator motor DC. Sistem akan menerima masukan dari dua sensor, yaitu sensor EMG dan sensor tekanan. Sensor EMG berfungsi mengukur kinerja otot kaki sedangkan sensor tekanan berfungsi mengukur tekanan yang diberikan pada bagian telapak kaki. 2 sinyal masukan ini akan diolah oleh mikrokontroller dengan menggunakan metode ANN. Sebelumnya, data yang didapatkan pada tahap pengujian awal akan digunakan untuk melatih sistem ANN untuk mendapatkan nilai *weight* dan *bias* yang akan digunakan pada mikrokontroller. Pelatihan pada ANN dilakukan dengan menggunakan metode *backpropagation*. Setelah tahap pelatihan selesai, nilai *weight* dan *bias* tersebut digunakan pada mikrokontroller untuk menentukan nilai sinyal PWM yang akan diberikan pada *driver* motor. *Driver* motor ini kemudian akan menggerakkan motor DC berdasarkan nilai PWM yang diterima dari mikrokontroller.



Gambar 3.1 Diagram Blok Sistem

Perangkat keras yang digunakan pada sistem meliputi:

1. Sensor EMG
Sensor ini digunakan untuk mengukur tingkat aktivasi otot kaki ketika melakukan gerakan mengayuh.
2. Sensor Tekanan
Sensor ini digunakan untuk mengukur besarnya tekanan yang diberikan pada FSR (*Force Sensitive Resistor*). Sensor akan menghasilkan tegangan yang berbanding lurus dengan besarnya tekanan yang diberikan pada FSR.
3. Arduino Nano
Perangkat mikrokontroler yang digunakan pada sistem ialah Arduino Nano. Arduino berfungsi untuk mengolah sinyal masukan dari dua sensor dan menghasilkan keluaran sesuai dengan perhitungan dari ANN.
4. Motor *Driver*
Perangkat *Driver* yang digunakan yaitu modul L298N. Driver ini berfungsi menggerakkan motor DC sesuai dengan nilai sinyal PWM yang diterima dari Arduino.
5. Motor DC
Motor DC yang digunakan adalah tipe motor DC 775. Motor ini berfungsi untuk menggerakkan roda belakang sepeda untuk membantu pengguna.

Perangkat lunak yang digunakan pada sistem meliputi:

1. Perangkat lunak pada komputer
Perangkat lunak yang digunakan yaitu program ANN yang digunakan untuk melatih sistem agar mampu mengenali dan membedakan tingkat beban pada pengguna dengan beberapa data yang digunakan sebagai *training dataset*. Metode pelatihan yang digunakan merupakan metode *backpropagation*. Program ditulis menggunakan bahasa C++ pada Visual Studio.
2. Perangkat lunak pada Arduino
Perangkat lunak pada Arduino digunakan untuk membaca masukan dari dua sensor kemudian mengolah data tersebut dengan metode *feedforward propagation*, kemudian mengubah

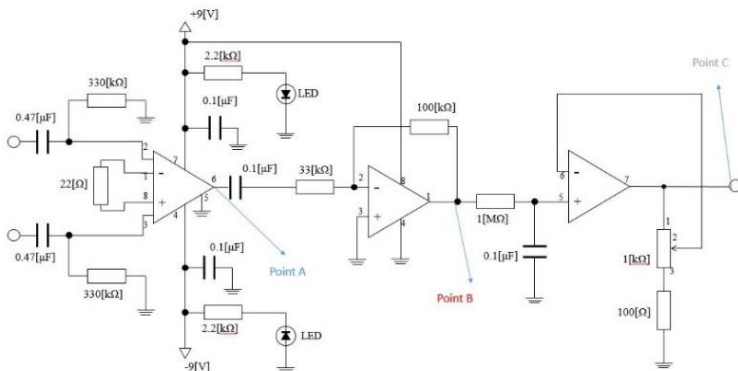
nilai yang dihasilkan melalui metode kontrol proporsional menjadi keluaran sinyal kontrol motor listrik, yaitu PWM.

3.2 Perancangan Perangkat Keras

Perangkat keras yang dirancang meliputi sensor EMG dan sensor tekanan.

3.2.1 Perancangan Sensor EMG

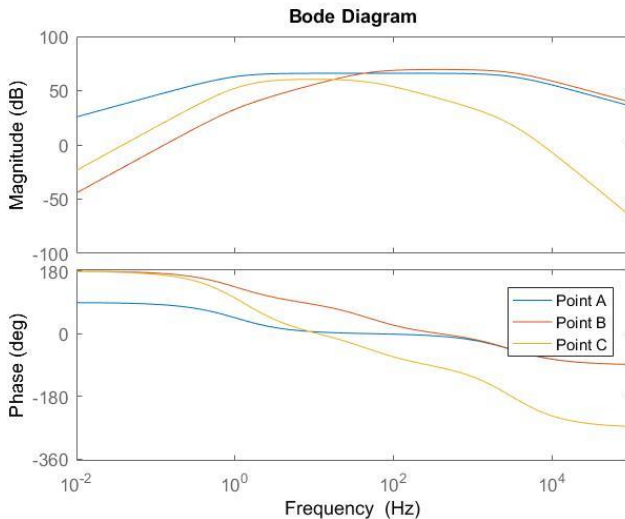
Rangkaian sensor EMG digunakan untuk mengetahui tingkat kinerja otot yang sedang diukur. Untuk mengukur sinyal EMG, digunakan rangkaian penguat diferensial. Konfigurasi penguat ini menggunakan dua titik masukan yang berbeda, dan rangkaian penguat akan mengurangkan dua sinyal dari dua titik masukan berbeda tersebut, sehingga komponen yang sama pada kedua sinyal hilang dan menyisakan komponen yang unik. Dengan konfigurasi ini, diharapkan sinyal EMG yang berasal dari titik pengukuran dapat dideteksi dengan optimal sedangkan *noise* yang tidak berasal dari titik pengukuran dapat dihilangkan. Namun, skema ini memerlukan penguat yang memiliki *Common Mode Rejection Ratio* (CMRR) yang tinggi. Semakin tinggi angka CMRR ini, maka penguat akan semakin mampu untuk menolak komponen yang sama pada sinyal yang akan diukur. Gambar 3.2 menunjukkan skematik rangkaian secara keseluruhan.



Gambar 3.2 Skematik Rangkaian Akuisisi Sinyal EMG

Pada rangkaian ini, IC untuk penguat instrumentasi menggunakan INA121 karena IC jenis ini memiliki CMRR yang cukup tinggi untuk mengurangi *noise* pada pengukuran sinyal, yaitu sebesar 106dB. Kemudian, penguat instrumentasi didesain dengan penguatan sebesar 2.273x. Sebelum sinyal masuk ke penguat, diberikan sebuah filter *high pass* dengan frekuensi *cutoff* sebesar 1Hz. Bagian ini ditunjukkan dengan poin A pada gambar 3.2

Berikutnya, sinyal akan masuk ke rangkaian *active high pass filter* 50Hz [13]. Pada titik ini, bagian negative dari sinyal akan dihilangkan. Kemudian, filter *low pass* 1Hz digunakan sebelum sinyal sampai ke tahap terakhir, yaitu *non-inverting amplifier* yang gainnya dapat diatur dengan menggunakan sebuah potensiometer. Selain itu, sinyal mengalami proses *smoothing* agar bentuk sinyal lebih teratur. Penguatan keseluruhan dari rangkaian ini dapat disesuaikan dari 7.568x hingga 83.250x.

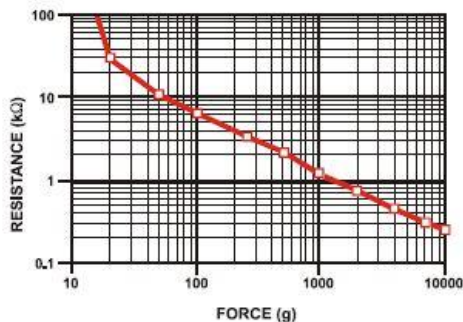


Gambar 3.3 Plot Diagram *Frequency vs Gain*

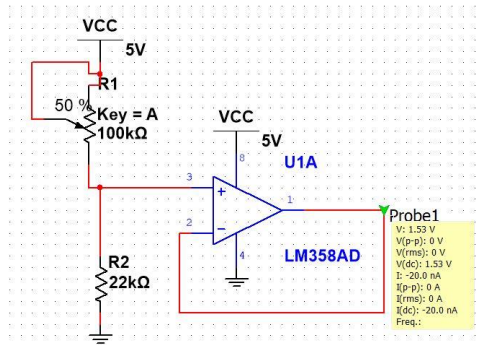
Gambar 3.3 menunjukkan hasil simulasi rentang frekuensi dari 3 titik pada rangkaian. Hasil keluaran akhir dari rangkaian, poin C, yang ditandai dengan warna kuning, memiliki puncak dari lebar frekuensi pada nilai 50Hz. Informasi yang penting pada sinyal EMG berada pada rentang frekuensi 50Hz-100Hz [14], sehingga rangkaian akan didesain dengan rentang frekuensi seperti 1 – 100 Hz agar informasi dari sinyal dapat ditangkap seutuhnya. Rangkaian notch filter 50Hz tidak digunakan karena dikhawatirkan akan menghilangkan komponen sinyal EMG yang penting pada frekuensi 50Hz[15] karena tidak ada filter notch ideal yang mampu menghilangkan komponen sinyal 50Hz saja tanpa menghilangkan komponen sinyal lain yang berdekatan.

3.2.2 Perancangan Sensor Tekanan

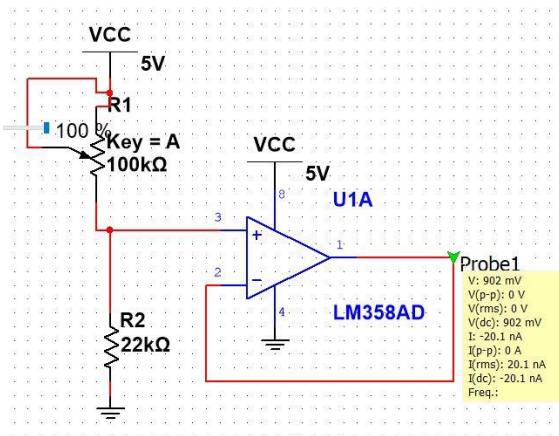
Sensor tekanan menggunakan sebuah *Force Sensitive Resistor* (FSR), yakni sebuah sensor resistif yang nilai resistansinya berubah sesuai dengan beban yang diberikan padanya. Gambar 3.4 menunjukkan grafik kerja dari FSR yang digunakan. Dari grafik kerja tersebut, dapat dilihat bahwa nilai resistansi FSR berbanding terbalik dengan besarnya tekanan yang diberikan padanya. Nilai hambatan pada saat tidak ada tekanan diatas 100kOhm, seolah bersifat *open circuit*, sedangkan pada saat tekanan maksimum, nilai resistansi sangat kecil, dibawah nilai 10Ohm.



Gambar 3.4 Grafik Kerja FSR

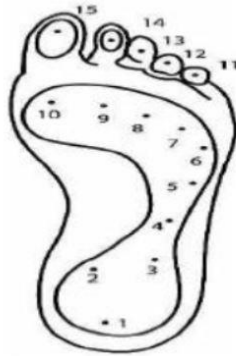


Gambar 3.5 Rangkaian Sensor Tekanan Kondisi Setengah Beban



Gambar 3.6 Rangkaian Sensor Tekanan Kondisi Tanpa Beban

Gambar 3.5 dan 3.6 menunjukkan rangkaian uji dari sensor tekanan menggunakan FSR. Dari kedua gambar tersebut, dapat dilihat bahwa rangkaian didesain sedemikian rupa agar nilai tegangan berbanding lurus dengan besar tekanan yang diberikan pada FSR. Sehingga, rentang tegangan keluaran berkisar dari 0.9 – 3.5V. Pada rangkaian ini catu daya yang digunakan sebesar 5VDC.



Gambar 3.7 Ilustrasi Titik Tekanan pada Telapak Kaki

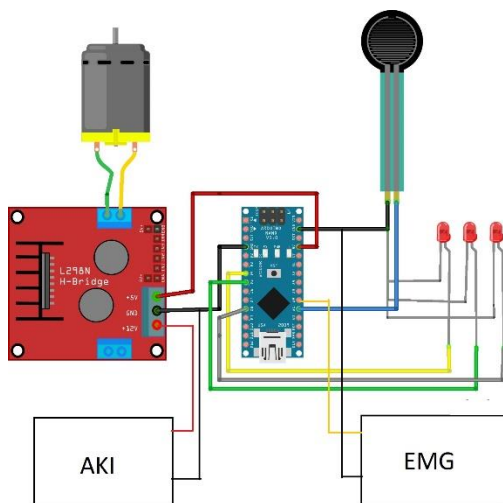
Pada riset ini, sensor tekanan ini diletakkan pada sebuah sandal dengan FSR ditempelkan pada bagian dalam sandal. Gambar 3.7 merupakan ilustrasi dari 15 titik tekanan pada telapak kaki manusia. Pada riset sebelumnya[16], digunakan 5 sensor untuk 5 titik tekanan pada kaki. Pada riset ini, untuk menentukan titik optimal peletakan FSR, dipilih tiga titik penempatan sensor. Data akan diambil pada masing-masing titik untuk melihat bentuk sinyal saat gerakan mengayuh. Titik yang akan digunakan yaitu titik nomor 1 (*posterior*), 5 (*medial*), dan 10 (*anterior*). Setelah melihat bentuk sinyal yang dihasilkan, akan dipilih satu dari ketiga titik uji ini untuk digunakan pada tahap penelitian berikutnya.

3.2.3 Arduino Nano

Pada tugas akhir ini, Arduino digunakan untuk membaca nilai tegangan dari sensor melalui ADC, mengolah nilai sensor dengan menjalankan algoritma *feedforward propagation* untuk menghasilkan nilai keluaran, kemudian menghasilkan sinyal PWM yang dimasukkan ke dalam *driver* motor berdasarkan sistem kontrol proporsional. Selain itu, terdapat 3 LED yang digunakan untuk indikator nilai keluaran ANN pada Arduino. Catu daya untuk Arduino menggunakan tegangan 5V dari modul L298. Tabel 3.1 menunjukkan penggunaan pin pada Arduino untuk pembacaan sensor, indikator keluaran, serta PWM.

Tabel 3.1 Penggunaan Pin Arduino

No.	Pin Arduino	Komponen	Keterangan
1	A1	Sensor FSR	Membaca tegangan dari sensor FSR
2	A2	Sensor EMG	Membaca tegangan dari sensor EMG
3	D3	PWM Motor DC	Menghasilkan sinyal PWM yang dimasukkan ke L298
4	D4	LED Indikator	Indikator kondisi keluaran
5	D5	LED Indikator	Indikator kondisi keluaran
6	D8	LED Indikator	Indikator kondisi keluaran
7	VIN	VIN	Menerima tegangan 5V dari modul L298



Gambar 3.8 Skema Pengkabelan Mikrokontroler

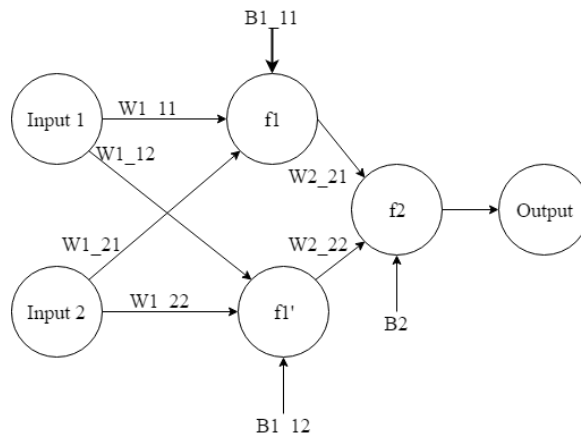
Gambar 3.8 menunjukkan skema pengkabelan dari mikrokontroler yang sudah dijelaskan pada tabel 3.1. Aki digunakan sebagai sumber untuk mikrokontroler serta motor DC. Namun, sensor EMG dan sensor tekanan kaki menggunakan catu daya baterai 9V.

3.3. Perancangan Perangkat Lunak

Perangkat lunak yang digunakan terbagi menjadi 2, yakni perangkat lunak pada Komputer dan pada Arduino. Perangkat lunak pada komputer digunakan untuk menjalankan program *Artificial Neural Network* (ANN) yang digunakan untuk melakukan proses pembelajaran terhadap sekumpulan data yang telah direkam sebelumnya. Hasil dari proses ini ialah nilai *weight* dan *bias* yang digunakan pada perangkat lunak di Arduino.

3.3.1 Perancangan Perangkat Lunak pada Komputer

Pada riset ini, ANN digunakan sebagai metode pembelajaran agar sistem mampu mengenali tingkat beban pada pengguna berdasarkan besarnya sinyal dari sensor EMG dan sensor tekanan pada pengguna. ANN merupakan metode pembelajaran yang cocok untuk digunakan karena memiliki tingkat kesuksesan yang tinggi[17] dengan jumlah kanal yang minimum[18]. Keluaran dari ANN yaitu sebuah nilai yang mengindikasikan tingkat aktivitas kaki subyek, yaitu dalam rentang nilai 0 hingga 1. Nilai nol menandakan aktivitas minimal dari kaki subyek dan tidak memerlukan bantuan motor sedangkan nilai 1 menandakan aktivitas maksimal pada kaki subyek sehingga motor akan dinyalakan dalam kecepatan maksimum.



Gambar 3.9 Diagram ANN yang akan Direalisasikan

Diagram dari ANN yang akan direalisasikan dapat dilihat pada gambar 3.9. Gambar tersebut menunjukkan sebuah ANN dengan 4 *layer*/lapisan yang terdiri dari 1 lapisan masukan, 2 lapisan tersembunyi, dan 1 lapisan keluaran. *Input* 1 merupakan sensor EMG, *Input* 2 merupakan sensor tekanan, dan *Output* merupakan nilai keluaran yang digunakan dalam proses selanjutnya. Nilai *weight* 1 terdiri dari sebuah matriks 2x2, nilai *bias* 1 terdiri dari matriks 1x2, nilai *weight* 2 terdiri dari matriks 2x1, dan nilai *bias* 2 terdiri dari matriks 1x1. Keempat parameter yang akan mengalami proses pembelajaran ini menentukan hasil keluaran ANN. Kemudian, terdapat 3 *neuron*, yang ditandai dengan f_1 , f_1' , dan f_2 pada gambar tersebut. Fungsi aktivasi yang digunakan pada f_1 dan f_1' merupakan fungsi log sigmoid sedangkan fungsi yang digunakan pada f_2 merupakan fungsi linier.

Struktur ANN pada Gambar 3.9 akan direalisasikan ke dalam sebuah program di komputer untuk melaksanakan fungsi pembelajaran sehingga proses pembelajaran berlangsung *offline* / terpisah dari mikrokontroler pada alat. Program ini berfungsi untuk melaksanakan fungsi pembelajaran setelah data-data dari subyek selama proses perekaman data telah sukses diakuisisi.

Program disusun menggunakan bahasa C++ dalam IDE Visual Studio. Model ANN yang digunakan seperti ditunjukkan pada gambar 3.8 ialah *Feedforward Multilayer Neural Network* dan metode pembelajaran menggunakan algoritma *Backward propagation*[18]. Dalam program ANN ini, terdapat beberapa tahapan dalam pengolahan data, yaitu:

1. Input Data

Pada tahap ini, pengguna memasukkan sekumpulan data yang akan digunakan dalam proses learning beserta sebuah *setpoint* atau target untuk tiap data tersebut. Sehingga, bila terdapat n data, maka jumlah *setpoint* yang harus dimasukkan juga sebesar n .

2. Normalisasi Data

Normalisasi dilakukan untuk menskalakan nilai masukan yang diberikan dalam suatu rentang yang didefinisikan sebelumnya. Hal ini bertujuan agar proses pembelajaran jaringan menjadi optimal. Proses normalisasi menggunakan nilai x_{min} dan x_{max} , yakni nilai maksimum dan minimum

dari keseluruhan data yang dimasukkan. Berikut merupakan rumus dalam menghitung nilai normalisasi:

$$norm = \frac{x - xmin}{xmax - xmin} \quad (3.1)$$

Dengan menggunakan rumus diatas, dengan x menunjukkan nilai sensor, nilai tersebut diskalakan kedalam skala antara nol dan satu.

3. *Feedforward Propagation*

Pada tahap ini, proses pembelajaran pada jaringan dimulai. Nilai masukan yang telah dinormalisasi akan diolah kedalam suatu jaringan *hidden layer* dengan menggunakan sebuah fungsi aktivasi. Fungsi aktivasi yang digunakan pada program ini adalah fungsi log sigmoid, karena fungsi log sigmoid ini merupakan fungsi yang mendekati cara kerja otak manusia[8].

4. Perhitungan Error

Pada tahap ini, setelah menentukan keluaran untuk tiap jumlah data yang dimasukkan, error dihitung dengan rumus sebagai berikut:

$$\text{Error} = \text{Setpoint} - \text{Output} \quad (3.2)$$

Kemudian, nilai error ini akan digunakan untuk melakukan fungsi pembelajaran *backpropagation*.

5. *Backward Propagation*

Metode pembelajaran *backpropagation* dilakukan untuk mengubah nilai parameter *weight* dan *bias* berdasarkan dari nilai error yang didapatkan dalam perhitungan sebelumnya. Dengan melakukan perubahan parameter *weight* dan *bias*, diharapkan pada iterasi pembelajaran berikutnya nilai error menjadi lebih kecil. Dalam perhitungan ini, terdapat beberapa parameter yang menentukan proses pembelajaran serta *update parameter* berikutnya:

$$s^M = -2 * F^M(n^M) * (t - a) \quad (3.3)$$

$$s^m = F^m(n^m) * (w^{m+1})^T * s^{m+1} \quad (3.4)$$

Rumus diatas berlaku untuk nilai $m = M-1, \dots, 2, 1$. Rumus diatas digunakan untuk menghitung nilai sensitivitas yang akan digunakan untuk *update parameter* nilai *weight* dan *bias*. Kemudian, terdapat rumus untuk *update parameter* dari *weight* dan *bias* sendiri, yakni:

$$w^m(k+1) = w^m(k) - \alpha * s^m(a^{m-1})^T \quad (3.5)$$

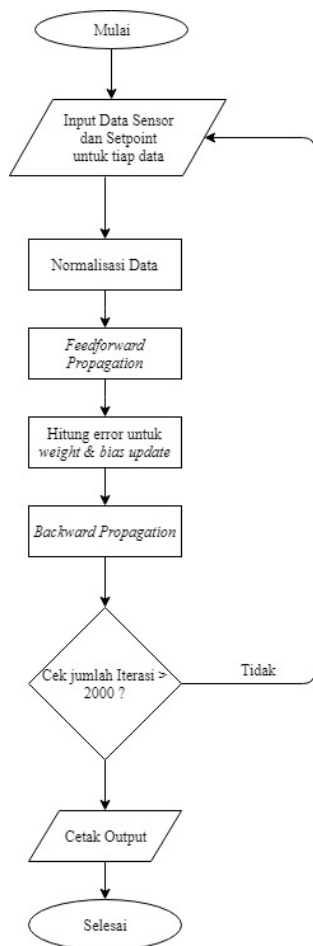
$$b^m(k+1) = b^m(k) - \alpha * s^m \quad (3.6)$$

Simbol alfa pada rumus diatas adalah *learning rate*. Jika semakin tinggi nilainya, nilai *weight* dan *bias* mengalami perubahan yang semakin cepat, namun dapat mengalami penurunan dalam akurasi hasil akhir yang didapatkan. Rumus-rumus diatas digunakan untuk merubah nilai *weight* dan *bias* agar hasil pembelajaran semakin mendekati target yang diinginkan.

6. Pengulangan Iterasi

Proses pembelajaran akan berlangsung sebanyak jumlah iterasi yang sudah dispesifikasikan sebelumnya, yaitu 2000 kali. Sehingga, proses pembelajaran secara keseluruhan sebanyak 2000 kali untuk tiap set data n yang dimasukkan kedalam program. Kemudian, program akan mengecek apakah jumlah iterasi yang dilakukan selama proses pembelajaran sudah mencapai 2000 kali. Jika belum, maka program akan mengulangi proses pembelajaran dan sebaliknya, jika sudah mencapai 2000 iterasi, maka program berhenti melakukan proses pembelajaran dan mencetak hasil akhir program beserta nilai *error* akhir yang didapatkan.

Secara keseluruhan, program menjalankan fungsi pembelajaran dari nilai sensor yang didapatkan agar sistem mampu mengenali tingkat aktivitas kaki subyek. Adapun nilai *weight* dan *bias* yang didapatkan dari program ini diimplementasikan dalam program ANN pada mikrokontroler. Gambar 3.10 menunjukkan alur dari program yang akan direalisasikan.



Gambar 3.10 Flowchart Program ANN

3.3.2 Perancangan Perangkat Lunak pada Arduino

Perangkat lunak pada Arduino didesain untuk membaca masukan dari 2 sensor, yaitu sensor EMG dan sensor tekanan, kemudian mengolah data tersebut menggunakan metode ANN, kemudian menghasilkan keluaran yang sesuai berdasarkan nilai *weight* dan *bias* yang didapatkan dari program ANN pada Komputer. Perbedaan program ANN pada Arduino dan Komputer ialah bahwa pada Arduino hanya menggunakan metode *forward propagation* saja, sehingga proses pembelajaran hanya berjalan satu arah. Hal ini dikarenakan mikrokontroller tidak memiliki kemampuan yang cukup untuk memproses informasi dan melakukan pembelajaran sekaligus. Dalam program Arduino ini terdapat beberapa tahap pengolahan data, yaitu:

1. Pembacaan Data

Arduino akan membaca data dari 2 sensor, yakni sensor EMG dan sensor tekanan. Data dibaca melalui ADC pada Arduino untuk disimpan dalam variabel penampung.

2. Normalisasi Data

Data yang sudah dibaca akan melalui proses normalisasi untuk merubah data tersebut kedalam rentang nilai yang diinginkan. Pada program ini, rentang nilai yang dispesifikasi ialah 0 hingga 1. Berikut merupakan rumus yang digunakan:

$$norm = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (3.7)$$

3. *Feedforward Propagation*

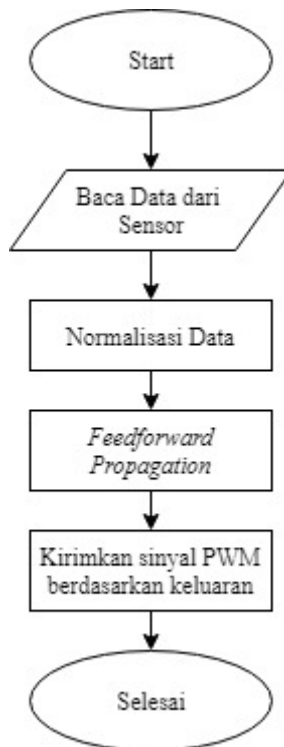
Pada tahap ini, data yang telah di-normalisasi akan diolah dengan menggunakan *weight* dan *bias* yang telah didapatkan dari program ANN pada komputer sehingga menghasilkan keluaran sesuai dengan yang ditentukan pada tahap pembelajaran.

4. Kirim Sinyal PWM

Pada tahap ini, mikrokontroller menghasilkan sinyal PWM keluaran sesuai dengan masukan sensor. Metode kontrol yang digunakan ialah metode kontrol proporsional, dengan kecepatan motor berubah berdasarkan nilai keluaran dari ANN. Semakin besar nilai keluaran ANN, maka kecepatan motor bertambah untuk mengompensasi beban pada pengguna. Sinyal PWM ini

dikirimkan ke *driver* motor L298N. Kemudian, modul L298N akan mengatur kecepatan pada motor listrik berdasarkan sinyal dari Arduino.

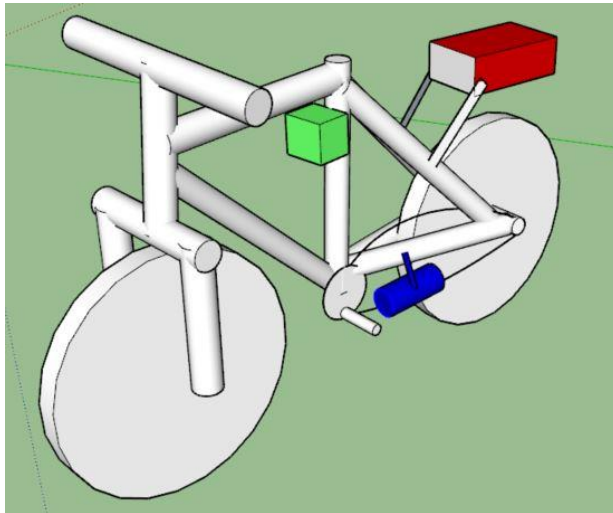
Secara keseluruhan, program ini berjalan secara terus menerus untuk membaca data dari kedua sensor yang digunakan, serta mengatur kecepatan motor listrik pada TAB berdasarkan nilai ANN dari Arduino. Gambar 3.11 menunjukkan alur kerja perangkat lunak pada Arduino.



Gambar 3.11 *Flowchart* Program Arduino

3.4 Perancangan Mekanik TAB

Gambar 3.12 menunjukkan desain mekanik dari sepeda TAB yang akan direalisasikan. Sebagian besar sepeda TAB ini menggunakan sepeda konvensional dengan beberapa penambahan komponen. Bagian berwarna hijau menunjukkan catu daya utama pada TAB, yaitu aki 12v untuk menggerakkan motor dan menyalakan rangkaian mikrokontroler. Bagian berwarna biru menunjukkan motor DC yang diletakkan di kerangka sepeda untuk menggerakkan ban belakang sepeda. Bagian berwarna merah yaitu kotak komponen yang meliputi sensor EMG, sensor tekanan, mikrokontroler, dan *driver* motor.



Gambar 3.12 Desain Mekanik TAB

BAB IV

PENGUJIAN DAN ANALISIS SISTEM

Bab ini membahas mengenai pengujian serta analisis kerja sistem *Torque Assisted Bicycle* yang telah dirancang pada bab sebelumnya. Pengujian dan analisis bab ini dibagi menjadi beberapa tahap, dengan membahas sub bagian dari keseluruhan sistem.

4.1 Realisasi Desain TAB

Gambar 4.1 menunjukkan realisasi desain sepeda TAB yang telah direncanakan pada bab sebelumnya. Sebuah penampang besi di atas ban belakang sebagai tempat menaruh kotak komponen yang berisikan rangkaian sensor dan mikrokontroler. Sebuah aki juga diletakkan di dekat sadel sepeda untuk memudahkan manajemen kabel yang dihubungkan dengan kotak komponen. Sebuah penyangga digunakan pada ban belakang sepeda agar ban tidak menyentuh permukaan tanah dan sepeda terletak pada posisi statis agar memudahkan tahap pengambilan data.



Gambar 4.1 Realisasi Desain Sepeda

4.2 Pengujian Perangkat Keras

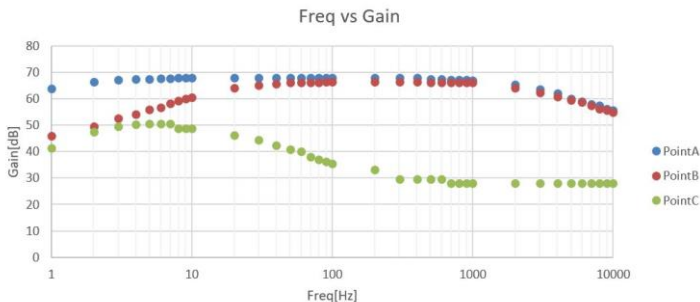
Pengujian perangkat keras dilakukan dengan cara menguji kedua perangkat keras yang digunakan, yakni sensor EMG dan sensor tekanan, kemudian menggabungkan keduanya untuk dibaca oleh mikrokontroler.

4.2.1 Pengujian Sensor EMG pada Tangan

Pengujian sensor EMG dilakukan dengan cara menguji respon sensor di tiap titik yang sudah dijelaskan pada bab sebelumnya ketika sensor digunakan pada otot tubuh. Pengujian tahap awal ini bertujuan untuk menguji apakah rangkaian sensor bekerja sesuai ekspektasi sebelum digunakan pada tahap pengambilan data pada gerakan *cycling*.

Gambar 4.2 menunjukkan hasil penguatan dalam satuan dB di rangkaian sensor EMG pada tiga titik uji dengan masukan yang frekuensinya diubah-ubah. Sinyal masukan merupakan sinyal sinusoidal dengan amplitudo 2mV dan frekuensi masukan bervariasi dari 1Hz hingga 10000Hz.

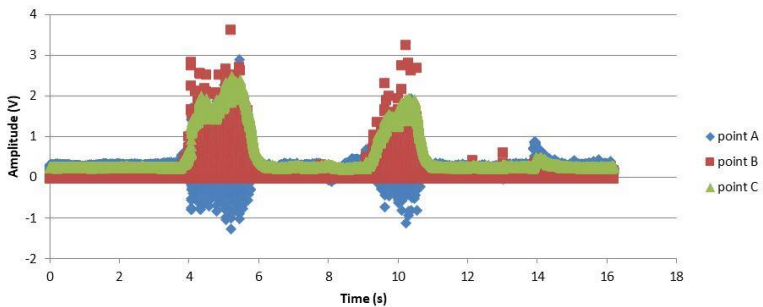
Poin A adalah titik keluaran dari penguat instrumentasi. Pengaruh *high pass* filter 1Hz terlihat pada grafik berwarna biru dengan penurunan nilai penguatan pada titik 1Hz. Poin B merupakan titik keluaran dari rangkaian *active high pass filter* 50Hz dengan penurunan nilai penguatan pada titik dibawah 50Hz. Poin C merupakan titik keluaran dari rangkaian penguat akhir, dan terdapat



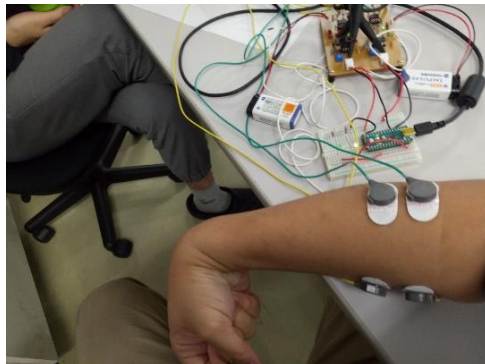
Gambar 4.2 Grafik Perubahan Frekuensi terhadap Gain Sensor EMG

sebuah *low pass filter* 1Hz. Sehingga, rentang frekuensi dari keseluruhan rangkaian ditunjukkan pada grafik poin C yang memiliki rentang frekuensi dominan pada nilai 1 hingga 90Hz.

Berikutnya, dilakukan pengujian pada subyek pada bagian tangan untuk memudahkan pemasangan elektroda dan pengambilan data. Pengujian dilakukan ketika subyek melakukan kontraksi otot tangan, sehingga terjadi perubahan pada amplitudo dari sinyal EMG. Sebelum pengambilan data, kulit subyek harus dibersihkan agar elektroda mampu menempel dengan sempurna. Berikut menggunakan hasil pengujian pada subyek tersebut:



Gambar 4.3 Respon Sensor terhadap Kontraksi Otot Tangan



Gambar 4.4 Peletakan Elektroda pada Tangan

Pengujian dilakukan dengan metode berikut:

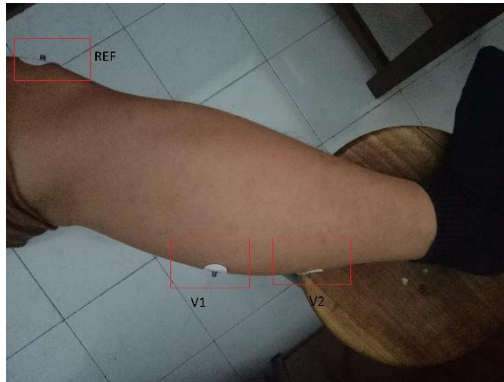
1. Subyek memasang elektroda ke otot tangan *flexor* setelah membersihkan permukaan kulit terlebih dahulu.
2. Setiap 5 detik, subyek melakukan gerakan fleksi dengan kondisi tubuh rileks dan duduk diam di kursi.
3. Sinyal gerakan direkam dan diolah kedalam bentuk grafis pada layar laptop.

Dari gambar 4.4 dapat dilihat respon sinyal EMG pada tiap titik pada rangkaian sensor. Poin A menunjukkan hasil sinyal EMG setelah melewati penguatan diferensial dan sinyal masih memiliki komponen negatif. Poin B menunjukkan sinyal yang sudah mengalami proses rektifikasi untuk menghilangkan komponen negatif sinyal. Poin C menunjukkan hasil sinyal terakhir setelah sinyal mengalami proses *smoothing* untuk menghaluskan bentuk sinyal. Sehingga, dari tiga titik yang ditinjau, dapat disimpulkan bahwa rangkaian akuisisi sinyal EMG sudah bekerja sebagaimana mestinya.

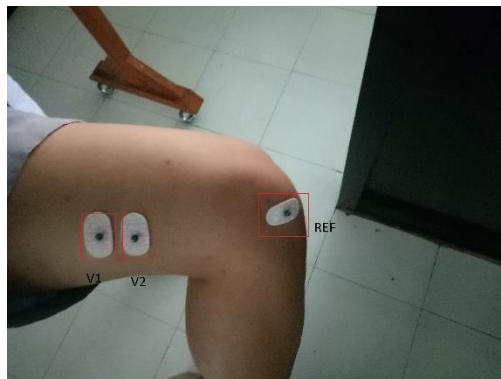
4.2.2 Pengujian Sensor EMG pada Gerak *Cycling*

Pengujian Sensor EMG dilakukan untuk mengetahui perbedaan amplitudo sinyal EMG dengan mengukur besarnya tegangan yang dihasilkan pada 3 kondisi beban yang telah ditentukan, yaitu beban minimal, setengah beban, dan beban maksimum yang diberikan dengan penggunaan tuas rem untuk roda belakang pada sepeda.

Beban minimal diberikan ketika tuas rem tidak ditekan, setengah beban diberikan ketika tuas rem ditekan sejauh 4cm dari posisi awalnya, dan beban maksimal diberikan ketika tuas rem ditekan sejauh 6,5cm dari posisi awalnya. Pengukuran dilakukan selama 10 detik dengan gerakan mengayuh pada kecepatan konstan di tiga kategori beban yang telah ditentukan sebelumnya. Dalam pengukuran ini digunakan 3 buah elektroda, dengan 2 elektroda untuk titik V1 dan V2, yaitu otot yang hendak dituju, dan 1 elektroda pada titik REF, yaitu titik pada anggota tubuh yang memiliki bagian tulang menonjol. Kemudian, hasil sinyal dibaca melalui arduino dan ditampilkan pada layar laptop.

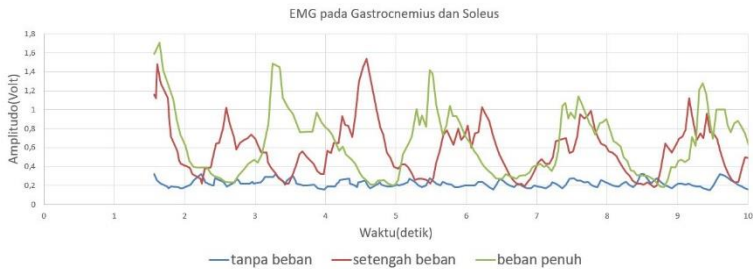


Gambar 4.5 Peletakan Elektroda pada Otot *Gastrocnemius* dan *Soleus*

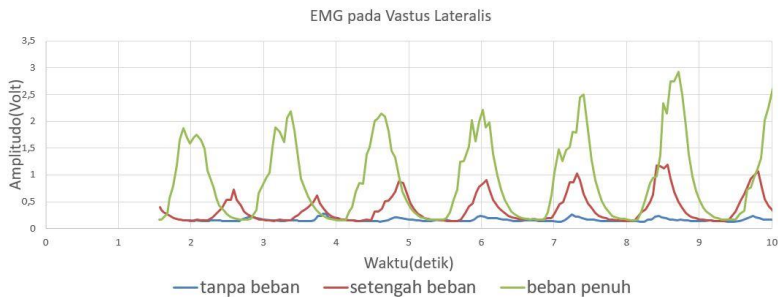


Gambar 4.6 Peletakan Elektroda pada Otot *Vastus Lateralis*

Gambar 4.5 dan 4.6 menunjukkan peletakan elektroda pada otot *Vastus Lateralis* dan *Gastrocnemius Soleus*. Titik V1 dan V2 merupakan titik akuisisi sinyal EMG yang akan dimasukkan ke rangkaian sedangkan titik REF merupakan titik referensi sinyal, yaitu bagian dari kaki yang keras (tulang menonjol) dan tidak terdapat otot. Sebelum melakukan pengujian dan perekaman sinyal, permukaan kulit terlebih dahulu dibersihkan untuk memastikan agar elektroda menempel dengan sempurna pada permukaan kulit agar tidak muncul *noise* yang mengganggu hasil pengukuran.



Gambar 4.7 Pengukuran EMG pada Otot *Gastrocnemius* dan *Soleus*



Gambar 4.8 Pengukuran EMG pada Otot *Vastus Lateralis*

Dua grafik diatas menunjukkan hasil pengukuran sinyal EMG pada 2 macam otot di kaki pada tiga tingkat beban yang berbeda. Grafik untuk kategori tanpa beban merupakan hasil pengukuran saat rem tidak ditekan. Grafik kategori setengah beban merupakan hasil pengukuran saat pengguna menekan tuas rem pada posisi tengah sedangkan grafik kategori beban penuh merupakan hasil ketika tuas rem ditekan sepenuhnya. Dari dua grafik diatas, dapat dilihat bahwa grafik EMG dari otot *Vastus Lateralis* menunjukkan hasil yang lebih optimal karena pola sinyal yang cukup konsisten selama sepuluh detik pengukuran.

4.2.3 Pengujian Sensor Tekanan pada Gerak *Cycling*



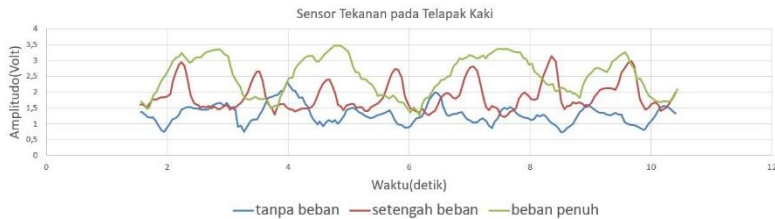
Gambar 4.9 Sensor Tekanan pada Sandal

Pengujian sensor tekanan dilakukan dengan mengukur besar tegangan yang dihasilkan pada tiga kondisi beban yang telah digunakan pada pengujian sebelumnya. Adapun kondisi subyek ketika akuisisi data sama dengan pengujian pada sub-bab 4.2.2.

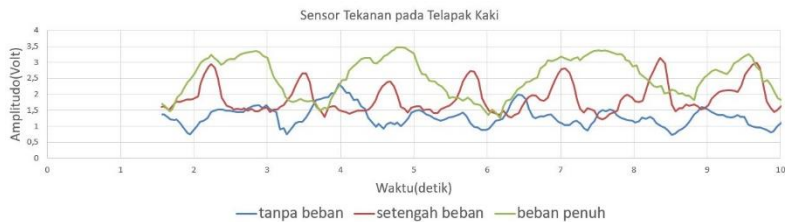
Pada pengujian ini akan dilakukan akuisisi data pada tiga titik yang berbeda di kaki, yaitu pada titik *medial* (telapak kaki), *anterior* (ibu jari), dan *posterior* (tumit). Gambar 4.9 menunjukkan sandal yang digunakan untuk akuisisi data dengan FSR diletakkan pada posisi tengah/*medial* telapak kaki. Pengukuran dilakukan pada titik tersebut dan dua titik lainnya.



Gambar 4.10 Pengukuran Sensor Tekanan pada titik *Anterior*



Gambar 4.11 Pengukuran Sensor Tekanan pada titik *Medial*

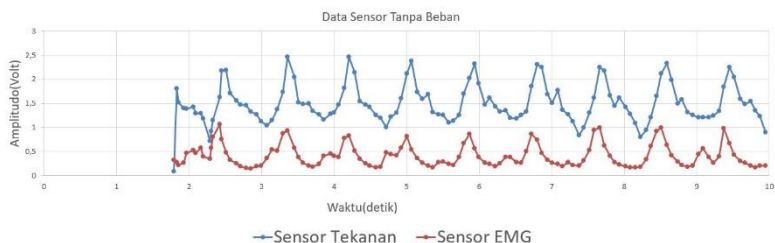


Gambar 4.12 Pengukuran Sensor Tekanan pada titik *Posterior*

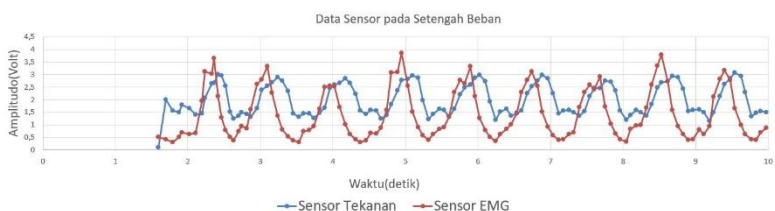
Dari ketiga hasil pengujian diatas, dapat dilihat bahwa karakteristik sinyal dari tiga titik pengujian saling berbeda satu dengan yang lain. Perubahan pada bentuk sinyal dan amplitudo dipengaruhi oleh penempatan *Force Sensitive Resistor* (FSR) pada titik di kaki. Dari hasil pengujian diatas, diputuskan bahwa untuk pengujian tahap berikutnya, FSR akan diletakkan pada titik *medial* karena sinyal yang dihasilkan pada titik ini memiliki bentuk yang cenderung konstan secara periodik serta memiliki amplitudo sinyal yang berbeda antara satu tingkat beban dengan yang lain.

4.2.4 Pengujian Sensor EMG dan Tekanan pada Gerak *Cycling*

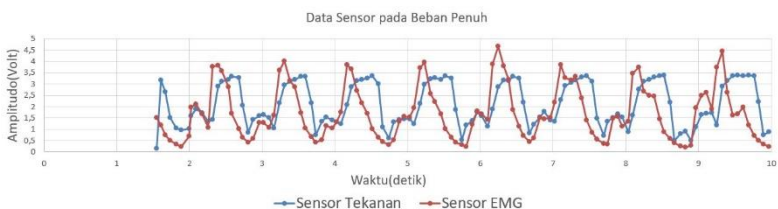
Pada tahap ini, dilakukan pengujian gerakan *cycling* dengan menggunakan sensor tekanan dan sensor EMG secara bersamaan. Kondisi pengujian dan definisi beban yang diberikan pada subyek masih sama dengan tahap sebelumnya, yaitu seperti pada pengujian di sub-bab 4.2.2. Peletakan sensor EMG pada otot *Vastus Lateralis* dan sensor tekanan pada telapak kaki. Berikut merupakan hasil pengukuran sensor:



Gambar 4.13 Pengukuran Sensor EMG dan Tekanan Kondisi Tanpa Beban



Gambar 4.14 Pengukuran Sensor EMG dan Tekanan Kondisi Setengah Beban

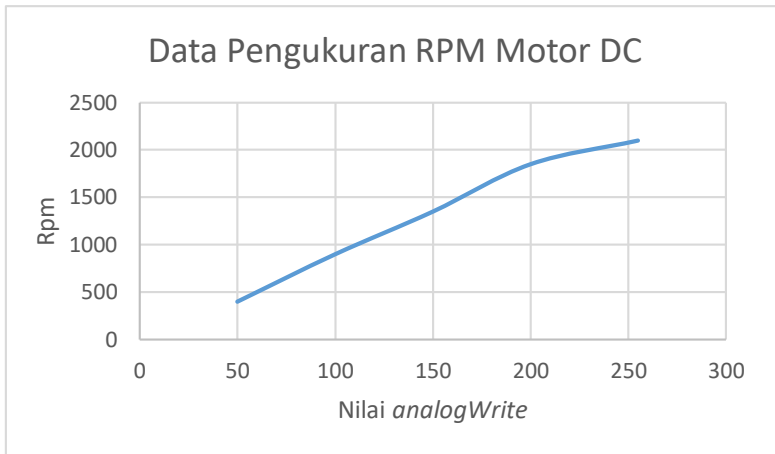


Gambar 4.15 Pengukuran Sensor EMG dan Tekanan Kondisi Beban Penuh

Dari gambar 4.13 hingga 4.15, dapat dilihat bahwa data sensor yang didapatkan pada tiga kondisi beban yang berbeda memiliki karakteristik bentuk dan amplitudo sinyal yang berbeda-beda. Bentuk sinyal secara keseluruhan cukup teratur walaupun pada kondisi setengah beban dan beban penuh, nilai amplitudo maksimum dari sinyal tidak memiliki perbedaan yang signifikan. Tahap pengujian ini dilakukan selama 3 hari yang berbeda untuk merekam data sensor pada tiga hari. Data-data ini akan digunakan untuk tahap berikutnya, yaitu tahap pembelajaran ANN.

4.2.5 Pengujian PWM pada Motor DC

Pengujian ini dilakukan untuk mengetahui pengaruh PWM dari perintah *analogWrite* yang diberikan dari Arduino pada motor DC yang terletak di TAB. Data yang diambil merupakan data rotasi motor tiap menitnya (rpm) dengan menggunakan alat ukur *tachometer* Yokogawa TM-300. Gambar 4.16 menunjukkan hasil pengukuran yang telah dilakukan. Dari grafik pengukuran, dapat dilihat bahwa nilai RPM pada motor DC berbanding lurus dengan peningkatan nilai PWM yang diberikan. Jika nilai rpm semakin besar, maka motor pun akan berputar semakin cepat. Dengan demikian, nilai PWM lebih besar akan diberikan pada tingkat beban yang besar, dan demikian pula sebaliknya.



Gambar 4.16 Grafik Perbandingan Nilai *analogWrite* dibandingkan RPM

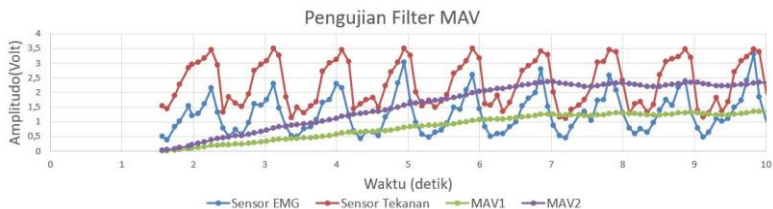
4.3 Pengujian Perangkat Lunak

Pada tahap ini dilakukan pengujian perangkat lunak pada sistem, yakni sistem pembelajaran ANN.

4.3.1 Pengujian Filter *Moving Average*

Pengujian filter *Moving Average* (MAV) ini dilakukan untuk mengetahui performa filter ketika digunakan pada sinyal EMG dan sinyal dari sensor tekanan pada saat gerakan mengayuh. Pada riset sebelumnya [19], telah digunakan filter MAV orde 12 untuk mengolah sinyal EMG untuk digunakan lebih lanjut. Filter MAV yang diimplementasikan pada mikrokontroler Arduino merupakan filter orde 60. Hal ini berarti filter secara kontinyu mencari nilai rata-rata dari 60 sampel data yang dibaca oleh mikrokontroler. Data yang digunakan terdiri dari 125 sampel data selama 10 detik perekaman data dari tahap sebelumnya. Tujuan implementasi filter ini ialah agar didapatkan sebuah representasi dari sinyal EMG dan sensor tekanan yang bersifat periodik serta mampu mengikuti perubahan pada amplitudo sinyal ketika terdapat perubahan beban yang diberikan pada subyek. Hasil pengujian filter dapat dilihat pada gambar

Gambar 4.17 menunjukkan hasil filter MAV pada sinyal EMG dan sinyal dari sensor tekanan. Dari grafik, filter membutuhkan waktu sekitar 6 detik sebelum mencapai nilai konstan. Adapun rentang nilai filter MAV ketika diterapkan pada sinyal EMG sebesar 1.3 dengan simpangan sebesar 0.1 dan untuk sensor tekanan sebesar 2.3 dengan simpangan sebesar 0.1. Filter ini akan diterapkan pada tiap data yang akan



Gambar 4.17 Pengujian Filter MAV

diolah dalam sistem pembelajaran, sehingga didapatkan suatu nilai yang merepresentasikan nilai sensor secara keseluruhan.

4.3.2 Pengujian Sistem Pembelajaran ANN

Program ANN ditulis dan di-*compile* pada laptop, sehingga proses pembelajaran berjalan secara terpisah dari mikrokontroller.

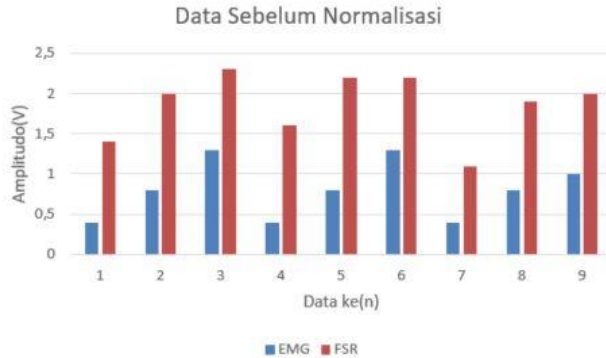
Kemudian, perlu diberikan suatu *setpoint* dalam pembelajaran ANN bagi tiap jenis data yang diberikan pada ANN. Data yang digunakan dalam proses pembelajaran terbagi menjadi 2 jenis dengan perbedaan terletak pada tempat peletakan elektroda untuk sensor EMG. Dari hasil pembelajaran, akan dihitung nilai kesalahan untuk menentukan penempatan elektroda yang optimal. Data A merupakan data sensor EMG yang digunakan di otot *Gastrocnemius* dan *Soleus* sedangkan data B merupakan data sensor EMG yang diletakkan pada *Vastus Lateralis*. Tiap set data terdiri dari 3 sub-data yang direkam pada tiga hari yang berbeda. Berikut merupakan data yang digunakan:

Tabel 4.1 Data Pembelajaran ANN A

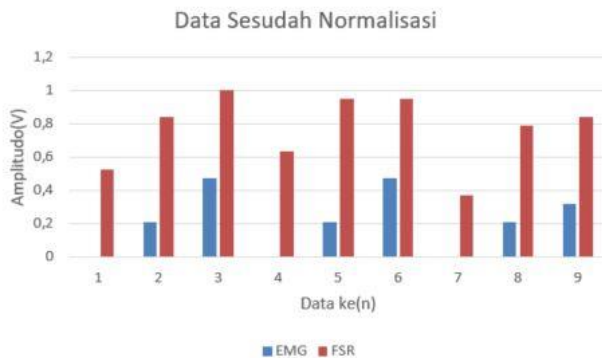
	Sensor EMG (V)	Sensor Tekanan (V)	<i>Setpoint</i>
Data 1	0,4	1,4	0
	0,8	2	0,5
	1,3	2,3	1
Data 2	0,4	1,6	0
	0,8	2,2	0,5
	1,3	2,2	1
Data 3	0,4	1,1	0
	0,8	1,9	0,5
	1	2	1

Tabel 4.2 Data Pembelajaran ANN B

	Sensor EMG (V)	Sensor Tekanan (V)	<i>Setpoint</i>
Data 1	0,6	1,1	0
	1,2	1,6	0,5
	1,5	2	1
Data 2	0,4	1,5	0
	1,4	2	0,5
	1,7	2	1
Data 3	0,2	1,5	0
	0,8	2	0,5
	1	2	1



Gambar 4.18 Data Pembelajaran sebelum Normalisasi



Gambar 4.19 Data Pembelajaran sesudah Normalisasi

Dari tabel 4.1 dan 4.2, secara keseluruhan terdapat 9 data yang terdiri dari data sensor EMG dan sensor tekanan beserta *setpoint* untuk tiap data. *Setpoint* 0 menandakan ‘tanpa beban’, 0,5 menandakan ‘setengah beban’, dan 1 menandakan ‘beban penuh’. Tiap data akan dimasukkan secara manual kedalam program beserta *setpoint* yang berkaitan.

Untuk meninjau proses pembelajaran, akan dilakukan pengecekan pada beberapa tahap dalam proses pembelajaran, dengan tahap pertama yaitu proses normalisasi data. Dari gambar 4.18 dan 4.19 dapat dilihat perbandingan dari 9 data pembelajaran ANN A sebelum dan

sesudah proses normalisasi dalam rentang 0 sampai 1. Untuk proses selanjutnya, data setelah normalisasi inilah yang akan digunakan dalam proses pembelajaran.

Setelah tahap pembelajaran berlangsung sebanyak iterasi yang telah dispesifikasikan, didapatkan nilai parameter *weight* dan *bias* yang sudah melalui proses pembelajaran. Nilai parameter ini akan digunakan pada perangkat lunak yang terdapat pada mikrokontroler di TAB. Grafik perubahan dari nilai parameter ini dapat dilihat pada gambar 4.20 untuk parameter *weight* 1 dan gambar 4.21 untuk parameter *bias* 1. Data yang digunakan pada gambar adalah data pembelajaran A.



Gambar 4.20 Perubahan Nilai *Weight* 1 terhadap Iterasi ANN



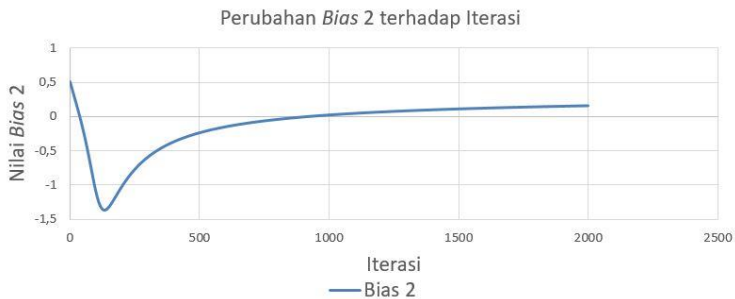
Gambar 4.21 Perubahan Nilai *Bias* 1 terhadap Iterasi ANN

Dari gambar 4.20, dapat dilihat bahwa laju perubahan nilai parameter W11 lebih cepat menuju nilai kestabilan sementara nilai parameter W12 masih terus meningkat. Adapun parameter *weight* 1 adalah matriks 2x2 dengan W11 adalah nilai matriks baris 1, kolom 1 sedangkan W12 adalah nilai matriks baris 1, kolom 2. Nilai yang ditunjukkan pada grafik hanyalah nilai W11 dan W12 karena nilai W21 sama dengan nilai W11 dan nilai W22 sama dengan nilai W12 sehingga tidak perlu adanya pengulangan grafik.

Gambar 4.21 menunjukkan bahwa parameter *Bias* 1 mengalami laju penurunan nilai hingga mencapai nilai negatif. *Bias* 1 adalah matriks 1x2 dengan nilai untuk B11 dan B21 adalah sama sehingga yang ditunjukkan pada gambar 4.21 hanya 1 garis saja.



Gambar 4.22 Perubahan Nilai *Weight* 2 terhadap Iterasi ANN



Gambar 4.23 Perubahan Nilai *Bias* 2 terhadap Iterasi ANN

Gambar 4.22 dan 4.23 menunjukkan laju perubahan nilai *weight* 2 dan *bias* 2. Laju perubahan nilai *weight* 2 mengalami peningkatan secara drastis pada iterasi awal kemudian mengalami peningkatan secara perlahan hingga iterasi akhir. Laju perubahan *bias* 2 mengalami penurunan secara besar pada iterasi awal, namun menunjukkan peningkatan hingga mencapai nilai stabil pada iterasi akhir. Dengan demikian, proses pembelajaran untuk nilai *weight* dan *bias* selesai.

Kemudian, untuk menentukan manakah yang lebih baik dari data pembelajaran A dan B, digunakan analisis *sum square error* (SSE). Berikut merupakan rumus yang digunakan untuk menghitung SSE:

$$SSE = \frac{1}{N} \sum_{i=0}^N (y - \hat{y})^2 \quad (4.1)$$

Variabel y merupakan keluaran program untuk data ke- i , \hat{y} merupakan *setpoint* untuk data ke- i , dan N merupakan jumlah data pembelajaran. SSE dihitung dengan cara menghitung *error* untuk tiap data pembelajaran, yaitu selisih antara keluaran program dengan *setpoint* untuk data tersebut. Setelah didapatkan *error* untuk tiap data, dihitung nilai kuadrat dari *error*, lalu dijumlahkan. Terakhir, hasil tersebut dibagi dengan banyak data yang dijumlahkan. Tabel 4.3 dan 4.4 merupakan tabel hasil perhitungan error untuk data pembelajaran A dan B.

Tabel 4.3 Data SSE A

	<i>Setpoint</i>	Hasil <i>Learning</i>	<i>Error</i>	<i>Square Error</i>	<i>Sum Square Error</i>
Data 1	0	0,229	0,229	0,052441	4,63%
	0,5	0,469	0,031	0,000961	
	1	1,23	0,23	0,0529	
Data 2	0	0,1727	0,1727	0,02982529	
	0,5	0,5332	0,0332	0,00110224	
	1	0,949	0,051	0,002601	
Data 3	0	0,082	0,082	0,006724	
	0,5	0,31	0,19	0,0361	
	1	0,516	0,484	0,234256	

Tabel 4.4 Data SSE B

	<i>Setpoint</i>	<i>Hasil Learnin g</i>	<i>Error</i>	<i>Square Error</i>	<i>Sum Square Error</i>
Data 1	0	-0,0026	0,0026	0,00000676	4,57%
	0,5	0,688	0,188	0,035344	
	1	0,93	0,07	0,0049	
Data 2	0	0,153	0,153	0,023409	
	0,5	0,852	0,352	0,123904	
	1	0,744	0,256	0,065536	
Data 3	0	-0,089	0,089	0,007921	
	0,5	0,607	0,107	0,011449	
	1	0,627	0,373	0,139129	

Dari dua tabel diatas, dapat dibandingkan bahwa data B menghasilkan nilai *error* yang lebih kecil dibandingkan dengan data A. Maka, untuk tahap pengujian akhir, otot yang digunakan untuk peletakan sensor EMG ialah otot *Vastus Lateralis*.

4.3.3 Pengujian Propagasi *Feedforward* pada Arduino

Pada tahap ini, nilai *weight* dan *bias* yang telah didapatkan pada tahap sebelumnya digunakan untuk propagasi *feedforward* pada Arduino. Untuk pengujian, digunakan data lain yang terpisah dari data yang digunakan untuk proses pembelajaran. Tabel 4.5 menunjukkan hasil dari pengujian ANN yang dilakukan pada Arduino dengan menggunakan nilai *weight* dan *bias* yang didapatkan dari tahap pembelajaran di laptop.

Tabel 4.5 Data Pengujian ANN pada Arduino

No	Sensor EMG	Sensor Tekanan	<i>Setpoint</i>	Hasil ANN	Error	SSE
1	0,7	1,3	0	0,09	0,09	8,93%
2	1,46	1,87	0,5	0,89	0,39	
3	1,51	2,1	1	1,1	0,1	
4	0,4	1,4	0	0	0	
5	1,5	2	0,5	1	0,5	
6	1,8	2,1	1	1,34	0,34	

Dari tabel tersebut, dapat dilihat tingkat kesalahan pengenalan sistem sebesar 8,93% dengan menggunakan perhitungan SSE. Selain itu, sistem mampu membedakan ada atau tidaknya beban pada pengguna, namun belum cukup sensitif dalam membedakan tingkat beban pada pengguna berdasarkan *setpoint* yang diberikan. Pada tahap pengujian berikutnya, akan digunakan program yang telah diuji pada tahap ini.

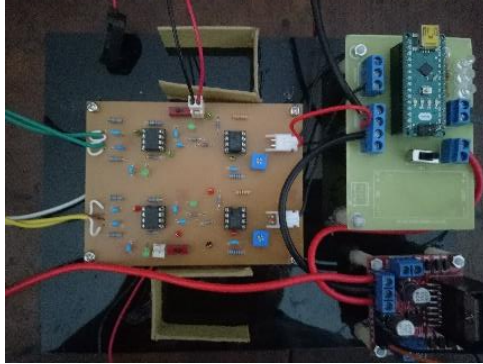
4.4 Pengujian Akhir pada Kondisi Statis

Pengujian keseluruhan alat pada kondisi statis dilakukan agar mengetahui performa alat sebelum dilakukan pengujian dinamis (pengujian riil pada permukaan lantai). Pengujian alat pada kondisi statis dilakukan dengan memberikan beban dari rem, kemudian mengamati perubahan yang terjadi pada indikator LED pada alat. Gambar 4.24 menunjukkan posisi subyek pada saat pengujian akhir sistem pada kondisi statis sedangkan gambar 4.25 menunjukkan wujud fisik alat yang diletakkan pada bagian belakang dari TAB.

Pada pengujian ini, terdapat 4 subyek dengan kondisi jasmani dan rohani yang sehat, kondisi otot prima (tidak melakukan aktivitas fisik yang berat menggunakan kaki sebelumnya), serta telah diinformasikan mengenai kriteria pengujian dari alat ini. Keempat subyek ini akan mengayuh sepeda selama 10 detik pada tiga kategori beban berbeda.



Gambar 4.24 Subyek pada Pengujian Statis



Gambar 4.25 Sistem Perangkat Keras Akhir

Kriteria pemberian beban pada subyek masih sama dengan pengujian pada sub-bab 4.2.2. Hasil keluaran ANN digunakan sebagai sinyal *error* untuk menentukan besarnya sinyal PWM yang akan menggerakkan motor DC. Evaluasi terhadap sistem dilakukan dengan melihat secara visual indikator lampu LED yang terdapat pada sistem. Terdapat 3 LED sebagai indikator tingkat beban yang diberikan pada subyek. LED 1 menyala jika hasil ANN berada dibawah 0,5 atau beban minimal, LED 2 menyala jika hasil ANN berada di antara 0,5 dan 1,0 atau beban setengah maksimum, dan LED 3 menyala jika hasil ANN lebih besar daripada 1 atau mengindikasikan beban maksimum.

Tabel 4.6 Data Pengujian Statis

Subyek	Beban	Indikator
Subyek 1	Minimal	LED 1
	Setengah	LED 2
	Maksimal	LED 3
Subyek 2	Minimal	LED 1
	Setengah	LED 2
	Maksimal	LED 3
Subyek 3	Minimal	LED 1
	Setengah	LED 1
	Maksimal	LED 3
Subyek 4	Minimal	LED 1
	Setengah	LED 1
	Maksimal	LED 3

Hasil pengujian statis pada 4 subyek ditunjukkan pada tabel 4.6. Dari tabel tersebut dapat dilihat bahwa pada pengujian yang telah dilakukan, sistem mampu mengenali perbedaan tingkat aktivitas kaki pada 2 subyek dengan tepat, namun sistem belum mengenali aktivitas kaki 2 subyek lain dengan tepat. Sehingga, tingkat kesuksesan deteksi sistem sebesar 83,33%. Adapun respon motor yang diberikan pada ban sesuai dengan kontrol proporsional yang diberikan sehingga motor berputar semakin cepat mengikuti bertambahnya beban yang diberikan pada subyek.

4.5 Pengujian Akhir pada Kondisi Dinamis

Pada tahap ini, dilakukan pengujian akhir pada permukaan lantai serta tanjakan dengan beberapa derajat kemiringan yang diatur untuk mengetahui respon alat. Dalam pengujian ini, digunakan 4 macam sudut kemiringan permukaan lantai, yaitu 0°, 10°, 15°, dan 20°. Pada pengujian ini ditentukan bahwa sudut 0° menandakan beban minimum, sudut 10° dan 15° menandakan beban setengah, dan sudut 20° menandakan beban maksimum. Hasil dari pengujian ini ditunjukkan pada Tabel 4.7. Dari tabel tersebut, dapat dilihat data hasil pengujian dinamis pada permukaan rata serta tanjakan dengan tiga sudut kemiringan. Pengujian dilakukan pada tiga sesi berbeda. Respon Hari I menunjukkan respon alat pada hari pertama pengujian. Respon Hari II menunjukkan respon alat pada hari kedua pengujian, sedangkan Respon Hari III menunjukkan respon alat pada hari ketiga pengujian. Angka 1 pada tabel menunjukkan indikator LED untuk kondisi beban minimal, angka 2 adalah indikator LED kondisi setengah beban, dan angka 3 menunjukkan indikator LED kondisi beban maksimal.

Tabel 4.7 Data Pengujian Dinamis

Derajat Kemiringan (°)	Respon Hari I	Respon Hari II	Respon Hari III
0	1	1	1
10	3	3	2
15	2	3	2
20	3	3	3

Secara keseluruhan, dapat dilihat bahwa sistem telah mampu membedakan beban yang diberikan pada pengguna berdasarkan tingkat kemiringan permukaan yang diujikan. Hasil pengujian pada tingkat kemiringan 0° dan 20° menunjukkan hasil yang konsisten dalam tiga kali pengujian. Namun, hasil pengujian pada tingkat kemiringan 10° dan 15° menunjukkan hasil yang tak konsisten karena indikator LED yang menyala tak konsisten. Adapun pada kedua tingkat kemiringan ini, indikator yang seharusnya menyala ialah indikator LED 2, karena beban yang diberikan masih kurang dari beban pada tingkat kemiringan 20° . Kesalahan ini dapat disebabkan beberapa faktor, seperti kualitas elektroda yang digunakan, kondisi kulit pada titik pengukuran, dan kondisi fisik subyek pada saat pengukuran. Sehingga, secara keseluruhan, dapat ditarik kesimpulan bahwa dari tiga kali pengujian pada empat titik kemiringan yang berbeda, sistem mampu mendeteksi beban dengan tingkat akurasi 75%.

Halaman ini sengaja dikosongkan

BAB V

PENUTUP

5.1 Kesimpulan

Pada riset ini, sebuah purwarupa *Torque Assisted Bicycle* telah diwujudkan dengan menggunakan sensor EMG dan sensor tekanan sebagai piranti masukan, kemudian mikrokontroler Arduino untuk mengolah masukan tersebut. Sensor EMG yang digunakan memiliki rentang frekuensi 1-100Hz dan penguatan sebesar 7.568x hingga 83.250x. Penempatan elektroda EMG yang optimal terletak pada otot *vastus lateralis* sedangkan penempatan FSR yang optimal terletak pada bagian *medial* kaki. Perangkat lunak akhir pada Arduino memiliki tingkat kesalahan deteksi beban sebesar 8,93% pada pengujian menggunakan dua set data. Pada pengujian statis, sistem mampu mendeteksi beban pada empat subyek dengan tingkat ketelitian 83,33%, sedangkan pada pengujian dinamis di permukaan lantai sebanyak tiga kali, sistem mampu mendeteksi beban pada subyek dengan tingkat ketelitian rata-rata 75%.

5.2 Saran

Saran untuk riset berikutnya yaitu penggunaan motor listrik dengan torsi lebih besar sebagai penggerak pada roda sepeda, misalnya seperti motor *brushless* dengan spesifikasi torsi sekitar 70NM dengan sistem rantai yang dihubungkan dengan ban belakang sehingga bantuan gerakan lebih terasa. Kemudian, dapat digunakan sistem pengiriman sinyal nirkabel seperti *bluetooth* pada sensor tekanan di kaki sehingga tidak perlu menggunakan kabel yang berpotensi menyebabkan kendala pada alat.

Halaman ini sengaja dikosongkan

DAFTAR PUSTAKA

- [1] N. Fukushima dan Y. Fujimoto, "Estimation of pedaling torque for electric power-assisted bicycle on slope environment," 2017, hlm. 1682–1687.
- [2] M. Zahak, "Signal Acquisition Using Surface EMG and Circuit Design Considerations for Robotic Prosthesis," dalam *Computational Intelligence in Electromyography Analysis - A Perspective on Current Applications and Future Challenges*, G. R. Naik, Ed. InTech, 2012.
- [3] A. J. Gmerek, M. Davoodi, N. Meskin, dan F. Jaber, "An EMG signal processing system for control of an ankle-foot orthosis," 2017, hlm. 0444–0449.
- [4] T. Tokuyasu, S. Kushizaki, S. Matsumoto, dan T. Kitawaki, "Development of automatic positioning system for bicycle saddle based on lower limb's EMG signals during pedaling motion," 2013, hlm. 27–32.
- [5] F. Martini, *Fundamentals of Anatomy & Physiology*, Ninth edition. Pearson Education, Inc, 2012.
- [6] F. Budiman, M. A. Nursyeha, M. Rivai, dan . S., "Pengenalan Suara Burung Menggunakan Mel Frequency Cepstrum Coefficient dan Jaringan Syaraf Tiruan pada Sistem Pengusir Hama Burung," *J. Nas. Tek. ELEKTRO*, vol. 5, no. 1, Mar 2016.
- [7] V. S. Kartika, M. Rivai, dan D. Purwanto, "Spoiled meat classification using semiconductor gas sensors, image processing and neural network," 2018, hlm. 418–423.
- [8] H. Widiantara, M. Rivai, dan D. Purwanto, "Implementation Analog Neural Network for Electronic Nose using Field Programmable Analog Arrays (FPAA)," *Int. J. Electr. Comput. Eng. IJECE*, vol. 2, no. 6, Des 2012.
- [9] D. S. A. Pambudi, M. Rivai, dan A. Arifin, "Detection of organic solvent compounds using optical fiber interferometer array and neural network pattern recognition," 2018, hlm. 477–482.

- [10] M. Rivai, A. Arifin, dan E. I. Agustin, "Mixed vapour identification using partition column-QCMs and Artificial Neural Network," 2016, hlm. 172–177.
- [11] K. Ogata, *Modern control engineering*, 5th ed. Boston: Prentice-Hall, 2010.
- [12] S. W. Smith, *The scientist and engineer's guide to digital signal processing*. San Diego, Calif.: California Technical Pub., 1999.
- [13] A. Hanafusa dan T. Shimizu, "Development of EMG Measurement System Aimed to Control Assisted Device," *Trans. Jpn. Soc. Med. Biol. Eng.*, vol. 52, hlm. O-32-O-33, Agu 2014.
- [14] C. J. De Luca, "Surface Electromyography: Detection and Recording." DelSys Incorporated, 2002.
- [15] J. Wang, L. Tang, dan J. E. Bronlund, "Surface EMG Signal Amplification and Filtering," *Int. J. Comput. Appl.*, vol. 82, no. 1, hlm. 15–22, Nov 2013.
- [16] P. S. Malvade, A. K. Joshi, dan S. P. Madhe, "IoT based monitoring of foot pressure using FSR sensor," 2017, hlm. 0635–0639.
- [17] S. Shin, J. Jung, dan Y. T. Kim, "A study of an EMG-based authentication algorithm using an artificial neural network," dalam *2017 IEEE SENSORS*, 2017, hlm. 1–3.
- [18] C. Calderon-Cordova, C. Ramirez, V. Barros, P. A. Quezada-Sarmiento, dan L. Barba-Guaman, "EMG signal patterns recognition based on feedforward Artificial Neural Network applied to robotic prosthesis myoelectric control," 2016, hlm. 868–875.
- [19] D. M. Sales, P. C. Galinho, P. F. Miguel, T. V. Duarte, dan F. João, "Multimodal User Interfaces to Improve Social Integration of Elderly and Mobility Impaired," *Stud. Health Technol. Inform.*, hlm. 14–25, 2012.

LAMPIRAN A

1. Baris Program Pembelajaran ANN

```
#include <iostream>
#include "conio.h"
#include "math.h"
#include "Header.h"
#include "time.h"

using namespace std;
void main()
{
    srand (time(NULL));
    matrixIn();
    normal();
    weightbiasInit();
    for(it=1;it<=500;it++)
    {
        learning();
        cout << it << endl;
    }
    cout << counter3 << endl;
    cout << "OUTPUT : " << endl;
    for(counter=1;counter<=n;counter++)
    {
        cout << out[counter] << " ";
    }
    cout << endl;
    cout << "WEIGHT 1: " << endl << weight[1][1] << " "
    << weight[1][2] << endl << weight[2][1] << " " << weight[2][2]
    << endl;
    cout << "WEIGHT 2: " << endl << weight2[1][1] << "
    " << weight2[1][2] << endl;
    cout << "BIAS 1: " << endl << bias1[1][1] << endl <<
    bias1[2][1] << endl;
    cout << "BIAS 2: " << bias2[1][1] << endl;
    cout << "ERROR : " << sse << endl;
```

```

        _getch();
    }

    void matrixIn()
    {
        cout << "Banyak Data: " ;
        cin >> n ;
        for (baris = 1; baris <= 2; baris++)
        {
            for(kolom =1; kolom <= n; kolom++)
            {
                cout << "input data sensor " << baris
                << " ke " << kolom << " = " ;
                cin>>matriks[baris][kolom];
            }
        }
        cout << "Input Selesai" ;
        cout << endl;

        for (baris = 1; baris <= 2; baris++)
        {
            for (kolom =1; kolom<= n; kolom++)
            {
                cout<< matriks[baris][kolom] << " ";
            }
            cout << endl;
        }

        cout << "Input Setpoint: " << endl ;
        for (kolom = 1; kolom <= n; kolom++)
        {
            cout << "Input setpoint ke "<< kolom
            << " = ";
            cin >> setpoint[kolom];
        }
    }
}

```

```

void normal()
{
    mini = matriks[1][1]; //asumsi
    for (baris = 1; baris <= n; baris++)
    {
        for (kolom=1; kolom<= n; kolom++)
        {
            if (matriks[baris][kolom] >= maxi)
            {
                maxi=matriks[baris][kolom]; // nilai
maksimum
            }
            elseif(matriks[baris][kolom]<=ini)
            {
                mini=matriks[baris][kolom]; //nilai
minimum
            }
        }
    }

    cout << "nilai maksimum: " << maxi << endl << "nilai
minimum: " << mini << endl;

    for (baris = 1; baris <= n; baris++)
    {
        for (kolom=1; kolom<= n; kolom++)
        {
            temp[baris][kolom]
(matriks[baris][kolom]-mini)/(maxi-mini); //normalisasi
        }
    }
    cout << "hasil normalisasi: " << endl;
    for (baris = 1; baris <= 2; baris++)
    {
        for (kolom = 1; kolom <= n; kolom++)

```

```

        {
            matriks[baris][kolom]= temp[baris][kolom];
            cout<< matriks[baris][kolom] << " ";
        }
        cout << endl;
    }
}

```

```

void weightbiasInit()
{
    for (baris = 1;baris <= 2;baris++)
    {
        for (kolom = 1;kolom <= 2;kolom++)
        {
            weight[baris][kolom] = 0.5; //nilai acak antara 0 dan 1
        }
    }
    for (baris = 1;baris <= 1;baris++)
    {
        for (kolom = 1;kolom <= 2;kolom++)
        {
            weight2[baris][kolom] = 0.5; //nilai acak antara 0 dan 1
        }
    }

    for (baris = 1;baris <= 2;baris++)
    {
        for (kolom = 1;kolom <= 1;kolom++)
        {
            bias1[baris][kolom] = 1;
        }
    }

    for (baris = 1;baris <= 1;baris++)
    {
        for (kolom = 1;kolom <= 1;kolom++)
    }
}

```

```

        {
            bias2[baris][kolom] = 1;
        }
    }
    cout<< "nilai weight 1:" << endl;
    for (baris = 1;baris <= 2;baris++)
    {
        for (kolom = 1;kolom <= 2;kolom++)
        {
            cout << weight[baris][kolom] << " " ;
        }
        cout << endl;
    }

    cout<< "nilai weight 2:" << endl;
    for (baris = 1;baris <= 1;baris++)
    {
        for (kolom = 1;kolom <= 2;kolom++)
        {
            cout << weight2[baris][kolom] << " " ;
        }
        cout << endl;
    }

    cout<< "nilai bias 1:" << endl;
    for (baris = 1;baris <= 2;baris++)
    {
        for (kolom = 1;kolom <= 1;kolom++)
        {
            cout << bias1[baris][kolom] << " " ;
        }
        cout << endl;
    }

    cout<< "nilai bias 2:" << endl;
    for (baris = 1;baris <= 1;baris++)

```

```

    {
        for (kolom = 1;kolom <= 1;kolom++)
        {
            cout << bias2[baris][kolom] << " " ;
        }
        cout << endl;
    }
}

void learning()
{
    lr=0.1;
    counter3++;
    //cout << "LEARNING" << endl;
    for(kolom =1;kolom <= n;kolom++)
    {
        for (baris =1;baris<=2;baris++)
        {
            temp2[baris][kolom] =0;
            for (counter = 1; counter <= 2; counter++)
            {
                temp2[baris][kolom]+=weight[baris][counter]*
matriks[counter][kolom] ; //w * p
            }
            temp2[baris][kolom]+= bias1[baris][1]; //w*p + b
        }
        temp2[2][kolom] << endl;
        for (baris = 1; baris <= 2; baris++)
        {
            z1[baris][kolom]=1/(1+(1/exp(temp2[baris][kolom]))); //log
sigmoid
        }
        for (counter = 1; counter <= 2; counter++)
        {
            z2[kolom] += weight2[1][counter] * z1[counter][kolom] ; //w2
* p2

```



```

    }
    z2[kolom] += bias2[1][1]; // w2*p2 + b2
    out[kolom] = z2[kolom]; // purelin
    z2[kolom] = 0;
    sse = setpoint[kolom] - out[kolom] ;
    //SENSITIVITY
    s2 = -2*sse; // s2 = -2 * F'(n) * error

    s1[1][1]=(z1[1][kolom]*(1-
z1[1][kolom])*weight2[1][1]) * s2; //s1 = F'(n) * (w2)T * s2
    s1[2][1]=(z1[2][kolom]*(1-
z1[2][kolom])*weight2[1][2]) * s2;

    for(baris=1;baris<=2;baris++)
    {
        ztrans[kolom][baris] = z1[baris][kolom];
    }
    for(baris=1;baris<=2;baris++)
    {
        intrans[kolom][baris]=matriks[baris][kolom];
    }

ztrans[kolom][2] << endl;

//UPDATE PARAMETER
    for(counter=1;counter<=2;counter++)
    {
        weight2[1][counter] -= lr*s2*ztrans[kolom][counter] ;
    }
    bias2[1][1] -= lr * s2;
    for(counter2=1;counter2<=2;counter2++)
    {
        for(counter=1;counter<=2;counter++)
        {
            temp2[counter2][counter] = 0;
            temp2[counter2][counter]=s1[counter2][1]*intrans[kol
om][counter];

```

```

    }
    }
    for(counter2=1;counter2<=2;counter2++)
        //{
        //
        for(counter=1;counter<=2;counter++)
            // {
            //cout << temp2[counter2][counter] << " ";
            // }
            // cout << endl;
            //}
    for (baris = 1;baris <= 2;baris++)
    {
        for (counter = 1;counter <= 2;counter++)
        {
            weight[baris][counter] -= lr * temp2[baris][counter];
        }

        cout << "weight1 : " << endl;
        for (baris = 1;baris <= 2;baris++)
        {
            for (counter = 1;counter <= 2;counter++)
            {
                cout<< weight[baris][counter] << " ";
            }
            cout << endl;
        }
        for(baris=1;baris<=2;baris++)
        {
            bias1[baris][1] -= lr* s1[baris][kolom];
        }
    }
    endl;
    //}
}
}

```

2. Baris Program pada Arduino

```
#include<Wire.h>
float ch1;
float ch2;
int pin1=1;
int pin2=2;
float
newdata,emg,sensor1,sensor2,weight_1_11,weight_1_21,weigh
t_1_12,weight_1_22,bias1_11,bias1_21,weight_2_11,weight_2
_12,bias2,z11,z21,out,norm1,norm2;
float
lastdata,data1,data2,data3,data4,data5,data6,data7,data8,data9,d
ata10,data11,data12,data13,data14,data15,data16,data17,data18
,data19,data20,data21,data22,data23,data24,data25,data26,data
27,data28,data29,data30,data31,data32,data33,data34,data35,da
ta36,data37,data38,data39,data40,data41,data42,data43,data44,
data45,data46,data47,data48,data49,data50,data51,data52,data5
3,data54,data55,data56,data57,data58,data59,data60;
float
temp2,temp4,temp6,temp8,temp10,temp12,temp14,temp16,tem
p18,temp20,temp22,temp24,temp26,temp28,temp30,temp32,te
mp34,temp36,temp38,temp40,temp42,temp44,temp46,temp48,t
emp50,temp52,temp54,temp56,temp58,temp60;
float
lastdataf,dataf1,dataf2,dataf3,dataf4,dataf5,dataf6,dataf7,dataf8
,dataf9,dataf10,dataf11,dataf12,dataf13,dataf14,dataf15,dataf16
,dataf17,dataf18,dataf19,dataf20,dataf21,dataf22,dataf23,dataf2
4,dataf25,dataf26,dataf27,dataf28,dataf29,dataf30,dataf31,dataf
32,dataf33,dataf34,dataf35,dataf36,dataf37,dataf38,dataf39,dat
af40,dataf41,dataf42,dataf43,dataf44,dataf45,dataf46,dataf47,d
ataf48,dataf49,dataf50,dataf51,dataf52,dataf53,dataf54,dataf55,
dataf56,dataf57,dataf58,dataf59,dataf60;
float
tempf2,tempf4,tempf6,tempf8,tempf10,tempf12,tempf14,tempf
16,tempf18,tempf20,tempf22,tempf24,tempf26,tempf28,tempf3
```

```

0,tempf32,tempf34,tempf36,tempf38,tempf40,tempf42,tempf44
,tempf46,tempf48,tempf50,tempf52,tempf54,tempf56,tempf58,
tempf60;
float expo,movi,movif,xmax,xmin;
float weight = 0.05;
float set;

void setup() {
    // put your setup code here, to run once:
    Serial.begin(9600);
    lastdata = 0;
    lastdataf = 0;
    weight_1_11 = 1.6403;
    weight_1_21 = 1.6403;
    weight_1_12 = 1.51605;
    weight_1_22 = 1.51605;
    bias1_11 = -2.57374;
    bias1_21 = -2.57374;
    weight_2_11 = 1.89309;
    weight_2_12 = 1.89309;
    bias2 = -0.769528;
    xmin = 0.4;
    xmax = 2.3;
}

void loop() {
    // put your main code here, to run repeatedly:
    ch1 = analogRead(pin1);
    ch2 = analogRead(pin2);
    newdata = (ch1/1023)*5;
    emg = (ch2/1023)*5;

    data1=emg;
    dataf1=newdata;
    movi =
    (data1+data2+data3+data4+data5+data6+data7+data8+data9+d

```

```

ata10+data11+data12+data13+data14+data15+data16+data17+
data18+data19+data20+data21+data22+data23+data24+data25
+data26+data27+data28+data29+data30+data31+data32+data3
3+data34+data35+data36+data37+data38+data39+data40+data
41+data42+data43+data44+data45+data46+data47+data48+dat
a49+data50+data51+data52+data53+data54+data55+data56+da
ta57+data58+data59+data60)/60;
movif=
(dataf1+dataf2+dataf3+dataf4+dataf5+dataf6+dataf7+dataf8+d
ataf9+dataf10+dataf11+dataf12+dataf13+dataf14+dataf15+dat
af16+dataf17+dataf18+dataf19+dataf20+dataf21+dataf22+data
f23+dataf24+dataf25+dataf26+dataf27+dataf28+dataf29+dataf
30+dataf31+dataf32+dataf33+dataf34+dataf35+dataf36+dataf3
7+dataf38+dataf39+dataf40+dataf41+dataf42+dataf43+dataf44
+dataf45+dataf46+dataf47+dataf48+dataf49+dataf50+dataf51+
dataf52+dataf53+dataf54+dataf55+dataf56+dataf57+dataf58+d
ataf59+dataf60)/60;
sensor1 = movi;
sensor2 = movif;

norm1 = (sensor1-xmin)/(xmax-xmin);
norm2 = (sensor2-xmin)/(xmax-xmin);
z11 = 1/(1 + (1/exp((weight_1_11*norm1 +
weight_1_12*norm2)+bias1_11)));
z21 = 1/(1 + (1/exp((weight_1_21*norm1 +
weight_1_22*norm2)+bias1_21)));
out = (weight_2_11 * z11 + weight_2_12 * z21) + bias2;
out += 0.22;
Serial.print(emg);
Serial.print(" ");
Serial.print(newdata);
Serial.print(" ");
//Serial.print(movi);
//Serial.print(" ");
//Serial.println(movif);
Serial.println(out);

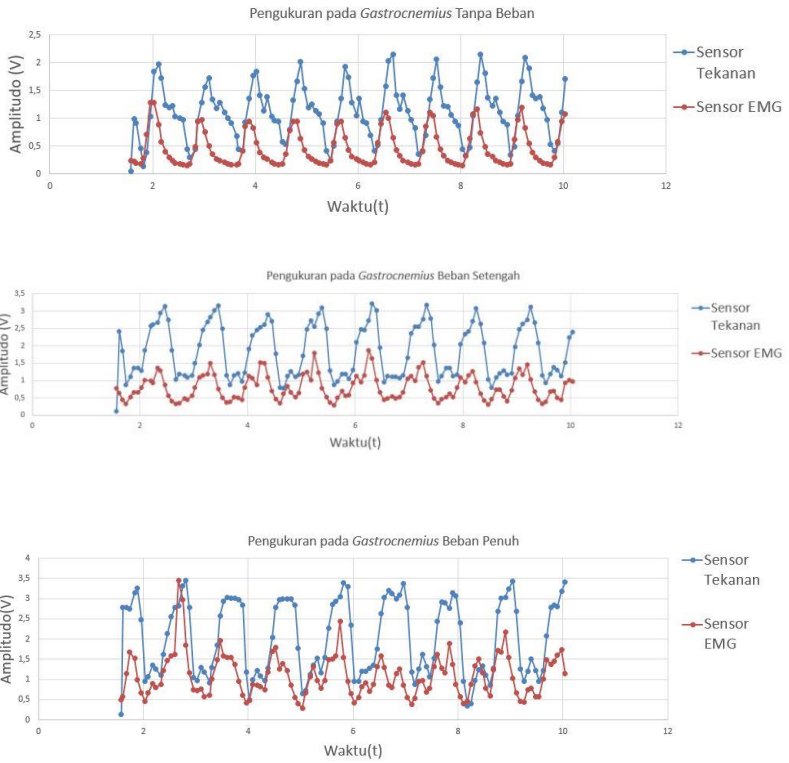
```

```

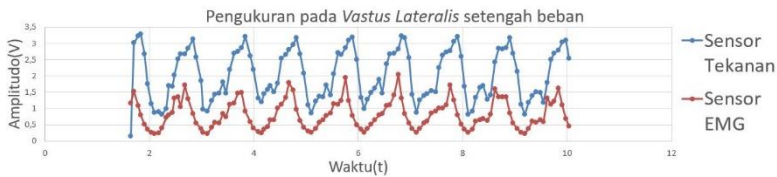
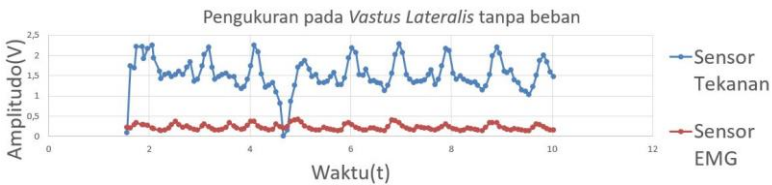
if (out >= 0 && out < 0.5)
{
    digitalWrite(4,HIGH);
    digitalWrite(7,LOW);
    digitalWrite(8,LOW);
    analogWrite(3,0);
}
else if (out >= 0.5 && out < 1)
{
    digitalWrite(4,LOW);
    digitalWrite(7,HIGH);
    digitalWrite(8,LOW);
    set = 300 * abs(out); //proporsional kp*error
    analogWrite(3,set);
}
else if (out >= 1)
{
    digitalWrite(4,LOW);
    digitalWrite(7,LOW);
    digitalWrite(8,HIGH);
    set = 300 * abs(out); //proporsional kp*error
    analogWrite(3,set);
}

```

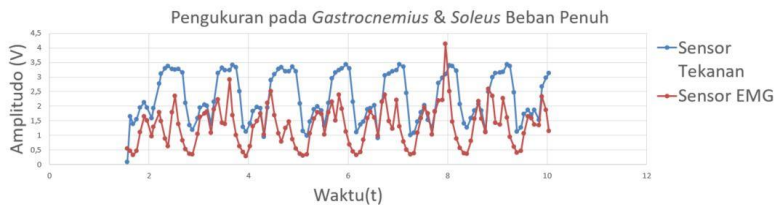
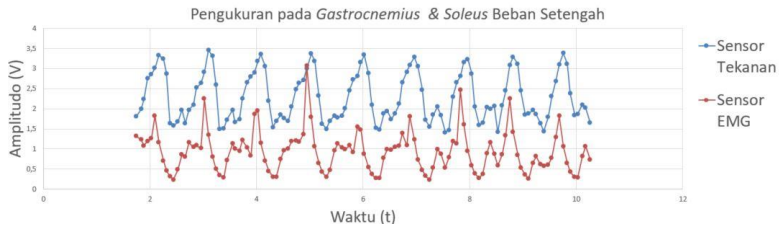
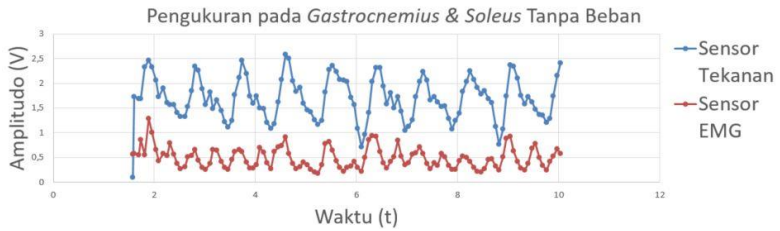
LAMPIRAN B
Hasil Pengukuran Sensor Selama 3 Hari
Sensor EMG pada *Gastrocnemius*, Hari 1



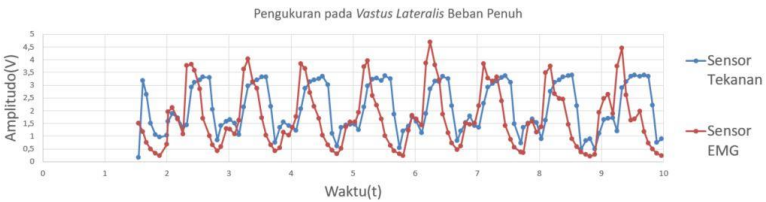
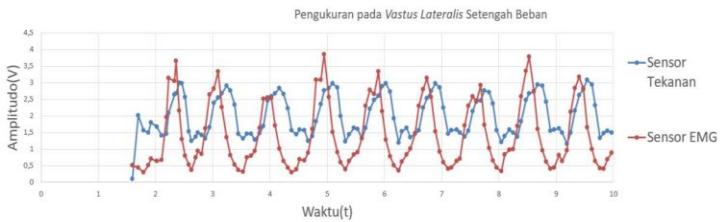
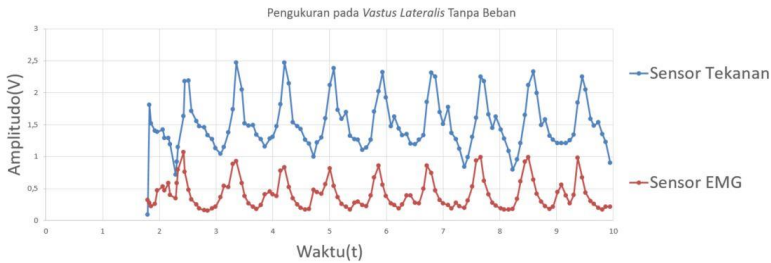
Sensor EMG pada *Vastus Lateralis*, Hari 1



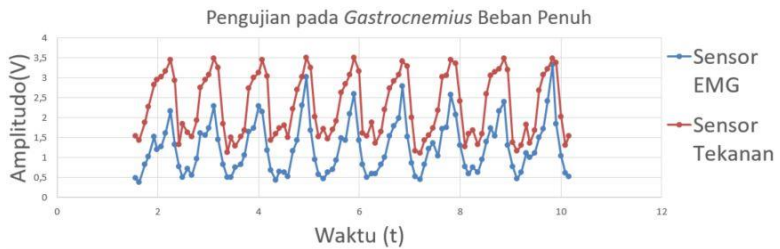
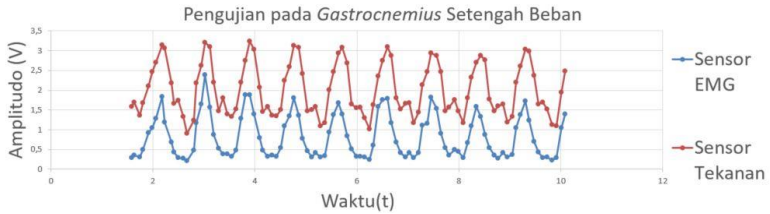
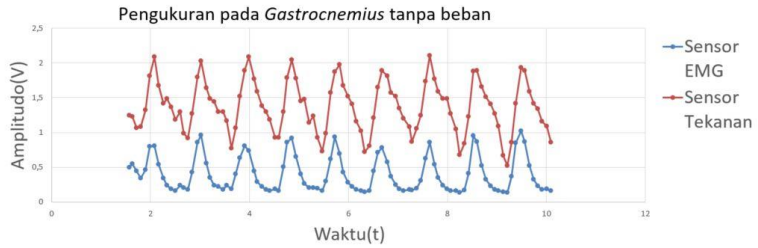
Sensor EMG pada *Gastrocnemius*, Hari 2



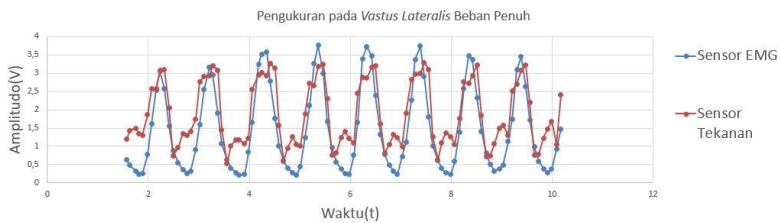
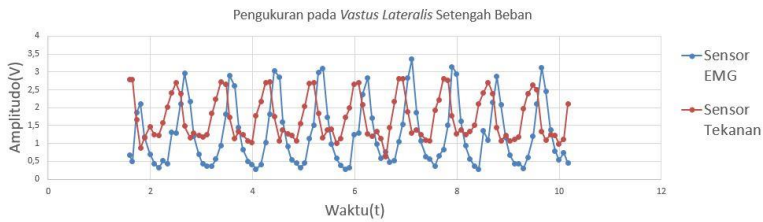
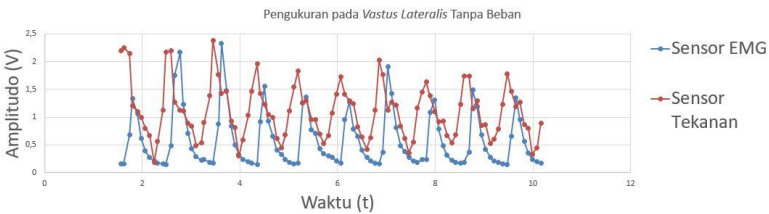
Sensor EMG pada *Vastus Lateralis*, Hari 2



Sensor EMG pada *Gastrocnemius*, Hari 3



Sensor EMG pada *Vastus Lateralis*, Hari 3



Perubahan pada *Weight* dan *Bias* setelah proses Pembelajaran pada data B





BIODATA PENULIS



Jeffrey Gunawan adalah mahasiswa program S1 Reguler angkatan 2014 di jurusan Teknik Elektro, Fakultas Teknologi Elektro, Institut Teknologi Sepuluh Nopember. Bidang yang ditekuni oleh penulis adalah elektronika dan penulis merupakan anggota laboratorium Elektronika Industri B402 di jurusan Teknik Elektro ITS. Penulis dilahirkan pada 19 April 1996 di kota Surabaya. Penulis menempuh pendidikan SMP di SMPK St. Carolus Surabaya pada tahun 2008-2011 dan SMA di SMAK St. Louis 1 Surabaya pada tahun 2011-2014. Cita-cita penulis ialah melanjutkan studi S2 di luar negeri dan mendirikan sebuah industri elektronik di bidang teknologi aplikatif.

Email: jeffreygunawan19@gmail.com